newslynx-sc-twitter

Release 0.0.1

August 31, 2015

Contents

CHAPTER 1

Installation

1.1 Production

To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

1.2 Development

If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

CHAPTER 2

Tests

Requires `nose`

```
$ make all_tests
```

CHAPTER 3

Documentation

Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

CHAPTER 4

Continuous Integration

Builds for `newslynx-sc-twitter` can be found on Travis

CHAPTER 5

Contributing

See the contributing guidelines.

CHAPTER 6

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`

        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.

    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

CHAPTER 7

Contents

7.1 newslynx-sc-twitter

7.1.1 Installation

Production

To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development

If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

7.1.2 Tests

Requires `nose`

```
$ make all_tests
```

7.1.3 Documentation

Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

7.1.4 Continuous Integration

Builds for `newslynx-sc-twitter` can be found on Travis

7.1.5 Contributing

See the contributing guidelines.

7.1.6 What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.

    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

7.1.7 Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests

Requires `nose`

```
$ make all_tests
```

Documentation

Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration

Builds for `newslynx-sc-twitter` can be found on Travis

Contributing

See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration

  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for newslynx-sc-twitter is hosted on Read The Docs.

It's generated via the following steps

- converts this file (README.md) into a ReStructured Text file, saving it to docs/index.rst
- runs newslynx sc-docs newslynx_sc_twitter -f rst to generate documentation for all the Sous Chefs in newslynx-sc-twitter and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for newslynx-sc-twitter can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

        * `make sous_chef_docs`
            - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
        * `make all_docs`:
            - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
        * `make view_docs`
            - Serves documentation at localhost:8000
        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000
    - `make register`:
      - Registers `newslynx-sc-twitter` on PyPI.
    - `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000
    - `make register`:
      - Registers `newslynx-sc-twitter` on PyPI.
    - `make distribute`:
      - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for newslynx-sc-twitter is hosted on Read The Docs.

It's generated via the following steps

- converts this file (README.md) into a ReStructured Text file, saving it to docs/index.rst
- runs newslynx sc-docs newslynx_sc_twitter -f rst to generate documentation for all the Sous Chefs in newslynx-sc-twitter and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for newslynx-sc-twitter can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000
    - `make register`:
      - Registers `newslynx-sc-twitter` on PyPI.
    - `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for newslynx-sc-twitter is hosted on Read The Docs.

It's generated via the following steps

- converts this file (README.md) into a ReStructured Text file, saving it to docs/index.rst
- runs newslynx sc-docs newslynx_sc_twitter -f rst to generate documentation for all the Sous Chefs in newslynx-sc-twitter and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for newslynx-sc-twitter can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000
    - `make register`:
      - Registers `newslynx-sc-twitter` on PyPI.
    - `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

        * `make sous_chef_docs`
            - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
        * `make all_docs`:
            - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
        * `make view_docs`
            - Serves documentation at localhost:8000
        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for newslynx-sc-twitter is hosted on Read The Docs.

It's generated via the following steps

- converts this file (README.md) into a ReStructured Text file, saving it to docs/index.rst
- runs newslynx sc-docs newslynx_sc_twitter -f rst to generate documentation for all the Sous Chefs in newslynx-sc-twitter and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for newslynx-sc-twitter can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000
    - `make register`:
      - Registers `newslynx-sc-twitter` on PyPI.
    - `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

        * `make sous_chef_docs`
            - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
        * `make all_docs`:
            - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
        * `make view_docs`
            - Serves documentation at localhost:8000
        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for newslynx-sc-twitter is hosted on Read The Docs.

It's generated via the following steps

- converts this file (README.md) into a ReStructured Text file, saving it to docs/index.rst
- runs newslynx sc-docs newslynx_sc_twitter -f rst to generate documentation for all the Sous Chefs in newslynx-sc-twitter and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for newslynx-sc-twitter can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000
    - `make register`:
      - Registers `newslynx-sc-twitter` on PyPI.
    - `make distribute`:
      - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for newslynx-sc-twitter is hosted on Read The Docs.

It's generated via the following steps

- converts this file (README.md) into a ReStructured Text file, saving it to docs/index.rst
- runs newslynx sc-docs newslynx_sc_twitter -f rst to generate documentation for all the Sous Chefs in newslynx-sc-twitter and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for newslynx-sc-twitter can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for newslynx-sc-twitter is hosted on Read The Docs.

It's generated via the following steps

- converts this file (README.md) into a ReStructured Text file, saving it to docs/index.rst
- runs newslynx sc-docs newslynx_sc_twitter -f rst to generate documentation for all the Sous Chefs in newslynx-sc-twitter and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for newslynx-sc-twitter can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000
    - `make register`:
      - Registers `newslynx-sc-twitter` on PyPI.
    - `make distribute`:
      - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for newslynx-sc-twitter is hosted on Read The Docs.

It's generated via the following steps

- converts this file (README.md) into a ReStructured Text file, saving it to docs/index.rst
- runs newslynx sc-docs newslynx_sc_twitter -f rst to generate documentation for all the Sous Chefs in newslynx-sc-twitter and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for newslynx-sc-twitter can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000

        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000
    - `make register`:
      - Registers `newslynx-sc-twitter` on PyPI.
    - `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for newslynx-sc-twitter is hosted on Read The Docs.

It's generated via the following steps

- converts this file (README.md) into a ReStructured Text file, saving it to docs/index.rst
- runs newslynx sc-docs newslynx_sc_twitter -f rst to generate documentation for all the Sous Chefs in newslynx-sc-twitter and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for newslynx-sc-twitter can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

        * `make sous_chef_docs`
            - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
        * `make all_docs`:
            - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
        * `make view_docs`
            - Serves documentation at localhost:8000
        * `make register`:
            - Registers `newslynx-sc-twitter` on PyPI.
        * `make distribute`:
            - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    - `make sous_chef_docs`
      - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    - `make all_docs`:
      - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    - `make view_docs`
      - Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:
      - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    - `make all_tests`:
      - Runs the tests.
    - `make readme`
      - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst

    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    - `make clean`:
      - Cleans out cruft from this directory.
    - `make install`:

        - Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
        - Runs the tests.
    * `make readme`
        - Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
        - Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
        - Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
        - Serves documentation at localhost:8000
    * `make register`:
        - Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
        - Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile

  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.

- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt

  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:
      · Publishes a new version of `newslynx-sc-twitter` to PyPI.
- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`

- tests
  - nose tests for newslynx-sc-twitter

Contents

newslynx-sc-twitter

Installation

Production To install newslynx-sc-twitter for an active installation of newslynx-core, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add newslynx-sc-twitter all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to newslynx-sc-twitter, do the following:

NOTE Will install a fresh version of newslynx via pip.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run newslynx-sc-twitter's Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires nose

```
$ make all_tests
```

Documentation Documentation for newslynx-sc-twitter is hosted on Read The Docs.

It's generated via the following steps

- converts this file (README.md) into a ReStructured Text file, saving it to docs/index.rst
- runs newslynx sc-docs newslynx_sc_twitter -f rst to generate documentation for all the Sous Chefs in newslynx-sc-twitter and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for newslynx-sc-twitter can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000
    * `make register`:
      · Registers `newslynx-sc-twitter` on PyPI.
    * `make distribute`:

        - Publishes a new version of `newslynx-sc-twitter` to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - `newslynx-sc-twitter`'s source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for `newslynx-sc-twitter`
- tests
  - `nose` tests for `newslynx-sc-twitter`

Contents

newslynx-sc-twitter

Installation

Production To install `newslynx-sc-twitter` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-twitter.git
```

To add `newslynx-sc-twitter` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development If you want to modify / add Sous Chefs to `newslynx-sc-twitter`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-twitter.git
$ cd newslynx-sc-twitter
$ pip install --editable .
```

You should now be able to run `newslynx-sc-twitter`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_twitter/say_my_name.yaml --myname='Brian Abelson'
```

Tests Requires `nose`

```
$ make all_tests
```

Documentation Documentation for `newslynx-sc-twitter` is hosted on Read The Docs.

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to docs/index.rst
- runs `newslynx sc-docs newslynx_sc_twitter -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-twitter` and saves the output to docs/sous-chefs.rst
- Builds Sphinx Documentation from these files.

Continuous Integration Builds for `newslynx-sc-twitter` can be found on Travis

Contributing See the contributing guidelines.

What's in this module ?

- README.md
  - This file
- VERSION
  - `newslynx-sc-twitter`'s source-of-truth version.
- requirements.txt
  - `newslynx-sc-twitter`'s python dependencies.
- MANIFEST.in
  - Specifications for which files to include in the PyPI distribution.
  - See the docs on this here.
- setup.py
  - Specification's for building `newslynx-sc-twitter`'s PyPI distribution.
- .travis.yml
  - Configurations for Travis Continuous Integration
  - You must activate this project on travis-ci.org for this to run on subsequent updates.
- Makefile
  - Helpers for managing `newslynx-sc-twitter`.
  - Includes:
    * `make clean`:
      · Cleans out cruft from this directory.
    * `make install`:
      · Installs `newslynx-sc-twitter`. Assumes that you're in a virtual environment.
    * `make all_tests`:
      · Runs the tests.
    * `make readme`
      · Converts this file to `.rst`, including a table of contents, and saves it to docs/index.rst
    * `make sous_chef_docs`
      · Programmtically generates Sous Chef documentation by running `newslynx sc-docs newslynx_sc_twitter/ --format=rst > docs/sous-chefs.rst`.
    * `make all_docs`:
      · Builds the sphinx docs for `newslynx-sc-twitter` by running the above two commands.
    * `make view_docs`
      · Serves documentation at localhost:8000

        * make register:
            · Registers newslynx-sc-twitter on PyPI.
        * make distribute:
            · Publishes a new version of newslynx-sc-twitter to PyPI.

- CONTRIBUTING.md
- newslynx_sc_twitter
  - newslynx-sc-twitter's source code and Sous Chef configuration files.
- docs
  - Sphnix documentation for newslynx-sc-twitter
- tests
  - nose tests for newslynx-sc-twitter

Contents

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module newslynx_sc_twitter.events.List.
- API Slug: twitter-list-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`

        * popular
        * both
    - Accepts inputs of type:
        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending

        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**

  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`

  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**

  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:

        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.

  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.

  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module newslynx_sc_twitter.events.SearchContentItemLinks.
- API Slug: twitter-search-content-item-links-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:

        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.

  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**

  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`

- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `searchstring`
    - `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    - `False`
  - Accepts inputs of type:
    - `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    - `pending`
    - `approved`
  - Accepts inputs of type:
    - `string`

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False

- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending

- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []

- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

- A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
- Should be rendered with a `search` form.
- Choose from:
- Accepts inputs of type:

  * `json`
- Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False

- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending

- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []

- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

- A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
- Should be rendered with a `search` form.
- Choose from:
- Accepts inputs of type:

  * `json`
- Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False

- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending

- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []

- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

- A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
- Should be rendered with a `search` form.
- Choose from:
- Accepts inputs of type:
  * `json`
- Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False

- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending

- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []

- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `searchstring`
    - `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    - `False`
  - Accepts inputs of type:
    - `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    - `pending`
    - `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

    - Accepts inputs of type:
        * `string`
    - Defaults to `pending`
- `set_event_title`
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a `text` form.
    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_description`
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a `paragraph` form.
    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `searchstring`
    - `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    - `False`
  - Accepts inputs of type:
    - `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    - `pending`
    - `approved`

    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    - `recent`
    - `popular`
    - `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

- A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
- Should be rendered with a `search` form.
- Choose from:
- Accepts inputs of type:
  * `json`
- Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    - `recent`
    - `popular`
    - `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    - `recent`
    - `popular`
    - `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    - `recent`
    - `popular`
    - `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

- A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
- Should be rendered with a `search` form.
- Choose from:
- Accepts inputs of type:
  * `json`
- Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

– Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False

- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending

- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None

- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []

- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    - `recent`
    - `popular`
    - `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * `string`
    - Defaults to `None`
- `set_event_tag_ids`
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a `checkbox` form.
    - Choose from:
    - Accepts inputs of type:
        * `string`
        * `numeric`
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a number form.
  - Accepts inputs of type:
    * numeric
    * nulltype
  - Defaults to 0
- search_query
  - **Required**
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * searchstring
    * nulltype
  - Defaults to None
- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string

- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * json
  - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to twitter-search-to-event and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    - `searchstring`
    - `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    - `False`
  - Accepts inputs of type:
    - `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    - `pending`
    - `approved`

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.

  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items

  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.

    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module newslynx_sc_twitter.events.Search.
- API Slug: twitter-search-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs **newslynx-sc-twitter** provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * string
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- search_query
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title

  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:

    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:

    * `string`

    * `numeric`

  - Defaults to `[]`

- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

    * `json`

  - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`

  - Defaults to `False`

- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:

```
    * json
```

  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:
    * `string`
  - Defaults to `recent`
  - More details on this option can be found here
- `search_query`
  - The query we use for additional filtration on text and urls.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`

- must_link
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a checkbox-single form.
  - Choose from:
    * False
  - Accepts inputs of type:
    * boolean
  - Defaults to False
- event_status
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a select form.
  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:

        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`

    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
    - Should be rendered with a checkbox form.
    - Choose from:
    - Accepts inputs of type:
        * string
        * numeric
    - Defaults to []
- set_event_content_items
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a search form.
    - Choose from:
    - Accepts inputs of type:
        * json
    - Defaults to []

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

Sous Chefs

newslynx-sc-twitter provides access to the following Sous Chefs

Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.

  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`

    - Defaults to `[]`
- `set_event_content_items`
    - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
    - Should be rendered with a `search` form.
    - Choose from:
    - Accepts inputs of type:
        * `json`
    - Defaults to `[]`

Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`

  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`
    * `both`
  - Accepts inputs of type:

        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string

  - Defaults to `pending`

- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.User`.
- API Slug: `twitter-user-to-event`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:

        * searchstring
        * nulltype
    - Defaults to None
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved
    - Accepts inputs of type:
        * string
    - Defaults to pending
- set_event_title
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_description
    - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
    - Should be rendered with a paragraph form.
    - Accepts inputs of type:
        * string
    - Defaults to None
- set_event_tag_ids
    - A list of Tag IDs or slugs to automatically apply to events created by this recipe.

  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module `newslynx_sc_twitter.metrics.OrgTimeseries`.
- API Slug: `twitter-user-to-org-timeseries`

Usage

Standalone Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics `twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`

7.2 Sous Chefs

newslynx-sc-twitter provides access to the following Sous Chefs

7.2.1 Twitter List To Event

- Extracts events from a twitter list.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.List`.
- API Slug: `twitter-list-to-event`

Usage

Standalone

Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_list_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes

Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_list_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-list-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development

Pass runtime options to `twitter-list-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_list_to_event.yaml --recipe=recipe.yaml
```

Options

In addition to default recipe options, `twitter-list-to-event` also accepts the following

- `list_owner_screen_name`

  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `list_slug`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
- `min_followers`
  - Filter out tweets from users with less than X followers.
  - **Required**
  - Should be rendered with a `number` form.
  - Accepts inputs of type:
    * `numeric`
    * `nulltype`
  - Defaults to `0`
- `search_query`
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `searchstring`
    * `nulltype`
  - Defaults to `None`
- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.

  - Choose from:
    * pending
    * approved
  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

7.2.2 Search For Links to Content Items

- This Sous Chef looks up all content items via the API and searches Twitter for tweets that links to the same URLs.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.SearchContentItemLinks`.
- API Slug: `twitter-search-content-item-links-to-event`

Usage

Standalone

Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes

Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_content_item_links_to_event.ya
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-content-item-links-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development

Pass runtime options to `twitter-search-content-item-links-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_content_item_links_to_event.yaml --recipe=recipe
```

Options

In addition to default recipe options, `twitter-search-content-item-links-to-event` also accepts the following

- `must_link`
  - Only create an event if there is a link to an existing content item.
  - **Required**
  - Should be rendered with a `checkbox-single` form.
  - Choose from:
    * `False`
  - Accepts inputs of type:
    * `boolean`
  - Defaults to `False`
- `event_status`
  - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
  - Should be rendered with a `select` form.
  - Choose from:
    * `pending`
    * `approved`
  - Accepts inputs of type:
    * `string`
  - Defaults to `pending`
- `set_event_title`
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - Defaults to `None`
- `set_event_description`
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a `paragraph` form.
  - Accepts inputs of type:
    * `string`

  - Defaults to `None`

- `set_event_tag_ids`
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a `checkbox` form.
  - Choose from:
  - Accepts inputs of type:
    * `string`
    * `numeric`
  - Defaults to `[]`
- `set_event_content_items`
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a `search` form.
  - Choose from:
  - Accepts inputs of type:
    * `json`
  - Defaults to `[]`

7.2.3 Twitter Search To Event

- Extracts events from a Twitter API query.
- This Sous Chef runs the python module `newslynx_sc_twitter.events.Search`.
- API Slug: `twitter-search-to-event`

Usage

Standalone

Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml --passthrough **op
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_search_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes

Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_search_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-search-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development

Pass runtime options to `twitter-search-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_search_to_event.yaml --recipe=recipe.yaml
```

Options

In addition to default recipe options, `twitter-search-to-event` also accepts the following

- `api_query`
  - The query to the Twitter API to return the initial batch of tweets.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`
  - More details on this option can be found here
- `result_type`
  - The type of tweets to return from the Twitter API.
  - **Required**
  - Should be rendered with a `select` form.
  - Choose from:
    * `recent`
    * `popular`

        * both
    - Accepts inputs of type:
        * string
    - Defaults to recent
    - More details on this option can be found here
- search_query
    - The query we use for additional filtration on text and urls.
    - **Required**
    - Should be rendered with a text form.
    - Accepts inputs of type:
        * searchstring
        * nulltype
    - Defaults to None
- min_followers
    - Filter out tweets from users with less than X followers.
    - **Required**
    - Should be rendered with a number form.
    - Accepts inputs of type:
        * numeric
        * nulltype
    - Defaults to 0
- must_link
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a checkbox-single form.
    - Choose from:
        * False
    - Accepts inputs of type:
        * boolean
    - Defaults to False
- event_status
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a select form.
    - Choose from:
        * pending
        * approved

  - Accepts inputs of type:
    * string
  - Defaults to pending
- set_event_title
  - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
  - Should be rendered with a text form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

7.2.4 Twitter User To Event

- Extracts events from a twitter user's timeline.
- This Sous Chef runs the python module newslynx_sc_twitter.events.User.
- API Slug: twitter-user-to-event

Usage

Standalone

Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml --passthrough **opti
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_event.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes

Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development

Pass runtime options to `twitter-user-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_event.yaml --recipe=recipe.yaml
```

Options

In addition to default recipe options, `twitter-user-to-event` also accepts the following

- `screen_name`

    - **Required**
    - Should be rendered with a `text` form.
    - Accepts inputs of type:
        * `string`
- `search_query`
    - **Required**
    - Should be rendered with a `text` form.
    - Accepts inputs of type:
        * `searchstring`
        * `nulltype`
    - Defaults to `None`
- `must_link`
    - Only create an event if there is a link to an existing content item.
    - **Required**
    - Should be rendered with a `checkbox-single` form.
    - Choose from:
        * `False`
    - Accepts inputs of type:
        * `boolean`
    - Defaults to `False`
- `event_status`
    - Set the status of the resulting events. Choose from pending and approved. Defaults to pending.
    - Should be rendered with a `select` form.
    - Choose from:
        * `pending`
        * `approved`
    - Accepts inputs of type:
        * `string`
    - Defaults to `pending`
- `set_event_title`
    - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
    - Should be rendered with a `text` form.
    - Accepts inputs of type:
        * `string`
    - Defaults to `None`

- set_event_description
  - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
  - Should be rendered with a paragraph form.
  - Accepts inputs of type:
    * string
  - Defaults to None
- set_event_tag_ids
  - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
  - Should be rendered with a checkbox form.
  - Choose from:
  - Accepts inputs of type:
    * string
    * numeric
  - Defaults to []
- set_event_content_items
  - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
  - Should be rendered with a search form.
  - Choose from:
  - Accepts inputs of type:
    * json
  - Defaults to []

7.2.5 Twitter User Timeseries Metrics

- Computes a timeseries of of metrics for one or more facebook pages.
- This Sous Chef runs the python module newslynx_sc_twitter.metrics.OrgTimeseries.
- API Slug: twitter-user-to-org-timeseries

Usage

Standalone

Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --passthrou
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes

Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=twitter-user-to-org-timeseries --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development

Pass runtime options to `twitter-user-to-org-timeseries` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_twitter/twitter_user_to_org_timeseries.yaml --recipe=recipe.yaml
```

Options

In addition to default recipe options, `twitter-user-to-org-timeseries` also accepts the following

- `screen_name`
  - The name of your twitter account.
  - **Required**
  - Should be rendered with a `text` form.
  - Accepts inputs of type:
    * `string`

Metrics

`twitter-user-to-org-timeseries` generates the following Metrics

- `twitter_followers`
  - Display name: `Twitter Followers`
  - Type: `cumulative`
  - Org Levels:
    * `timeseries`
    * `summary`