
newslynx-sc-rss

Release 0.0.1

Sep 27, 2017

Contents

1	Installation	1
1.1	Production	1
1.2	Development	1
2	Tests	3
3	Documentation	5
4	Continuous Integration	7
5	Contributing	9
6	What's in this module ?	11
7	Contents	13
7.1	Sous Chefs	13

CHAPTER 1

Installation

Production

To install `newslynx-sc-rss` for an active installation of `newslynx-core`, clone it and copy into `~/newslynx/sous-chefs`

```
$ git clone https://github.com/newslynx/newslynx-sc-rss.git
$ mv newslynx-sc-rss/ ~/newslynx/sous-chefs/
```

Now install it within the same virtual environment as `newslynx`:

```
$ cd ~/newslynx/sous-chefs/newslynx-sc-rss/
$ pip install .
```

... and if you're running `newslynx` as `sudo`

```
$ cd ~/newslynx/sous-chefs/newslynx-sc-rss/
$ sudo pip install .
```

Finally, run `newslynx sc-sync` to `newslynx-sc-rss`'s Sous Chefs for all organizations.

```
$ newslynx sc-sync
```

Development

If you want to modify / add Sous Chefs to `newslynx-sc-rss`, instal it in it's own virtual environment.

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

```
$ mkvirtualenv newslynx-sc-rss
$ git clone https://github.com/newslynx/newslynx-sc-rss.git
```

```
$ cd newslynx-sc-rss
$ pip install --editable .
```

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

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

CHAPTER 2

Tests

Requires nose

```
$ make all_tests
```


CHAPTER 3

Documentation

Documentation for `newslynx-sc-rss` 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_rss -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-rss` and saves the output to `docs/sous-chefs.rst`
- Builds Sphinx Documentation from these files.

CHAPTER 4

Continuous Integration

Builds for `newslinx-sc-rss` can be found on [Travis](#)

CHAPTER 5

Contributing

See the [contributing guidelines](#).

What's in this module ?

- `README.md`
 - This file
- `VERSION`
 - `newslinx-sc-rss`'s source-of-truth version.
- `requirements.txt`
 - `newslinx-sc-rss`'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 `newslinx-sc-rss`'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 `newslinx-sc-rss`.
 - Includes:
 - * `make clean`:
 - Cleans out cruft from this directory.
 - * `make install`:
 - Installs `newslinx-sc-rss`. 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
 - Programmatically generates `Sous Chef` documentation by running `newslynx sc-docs newslynx_sc_rss/ --format=rst > docs/sous-chefs.rst`.
- * make all_docs:
 - Builds the sphinx docs for `newslynx-sc-rss` by running the above two commands.
- * make view_docs
 - Serves documentation at `localhost:8000`
- * make register:
 - Registers `newslynx-sc-rss` on `PyPI`.
- * make distribute:
 - Publishes a new version of `newslynx-sc-rss` to `PyPI`.
- `CONTRIBUTING.md`
- `newslynx_sc_rss`
 - `newslynx-sc-rss`'s source code and `Sous Chef` configuration files.
- `docs`
 - Sphinx documentation for `newslynx-sc-rss`
- `tests`
 - nose tests for `newslynx-sc-rss`

Sous Chefs

newslynx-sc-rss provides access to the following Sous Chefs

Ingest Articles from an RSS Feed

- Extracts articles from an RSS Feed.
- This Sous Chef runs the python module `newslynx_sc_rss.feed.Article`.
- API Slug: `rss-feed-to-article`

API Usage

Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_rss/feed_to_article.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=rss-feed-to-article **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=rss-feed-to-article --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the `SousChef's load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development

Pass runtime options to `rss-feed-to-article` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc newslynx_sc_rss/feed_to_article.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc newslynx_sc_rss/feed_to_article.yaml --recipe=recipe.yaml
```

Options

In addition to default recipe options, `rss-feed-to-article` also accepts the following

- `feed_url`
 - **Required**
 - Should be rendered with a `text` form.
 - Accepts inputs of type:
 - * `url`

Google Alerts to Event

- Extracts events from a google alerts RSS feed.
- This Sous Chef runs the `python` module `newslynx_sc_rss.google_alerts.Events`.
- API Slug: `google-alert-to-event`

API Usage

Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_rss/google_alerts_to_event.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=google-alert-to-event **options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=google-alert-to-event --data=recipe.yaml
```

Save the outputted `id` of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development

Pass runtime options to `google-alert-to-event` and stream output. **NOTE** Will not execute the SousChef's `load` method.

```
$ newslynx sc newslynx_sc_rss/google_alerts_to_event.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc newslynx_sc_rss/google_alerts_to_event.yaml --recipe=recipe.yaml
```

Options

In addition to default recipe options, `google-alert-to-event` also accepts the following

- `feed_url`
 - An RSS Feed URL for a Google Alert
 - **Required**
 - Should be rendered with a `text` form.
 - Accepts inputs of type:
 - * `url`
 - More details on this option can be found [here](#)
- `must_link`
 - Only create an event if there is a link to an existing content item.
 - **Required**
 - Should be rendered with a `checkbox-single` form.
 - Choose from:
 - * `False`
 - Accepts inputs of type:
 - * `boolean`
 - Defaults to `False`
- `event_status`
 - Set the status of the resulting events. Choose from `pending` and `approved`. Defaults to `pending`.
 - Should be rendered with a `select` form.
 - Choose from:
 - * `pending`
 - * `approved`
 - Accepts inputs of type:
 - * `string`

- Defaults to `pending`
- `set_event_title`
 - Set's the title of the resulting events. This can be a python format string which has access to all of an event's top-level keys: IE: "Content from {authors} at {created}."
 - Should be rendered with a `text` form.
 - Accepts inputs of type:
 - * `string`
 - Defaults to `None`
- `set_event_description`
 - Set's the description of the output events. This can be a python format string which has access to all of an event's top-level keys: IE: "{title}."
 - Should be rendered with a `paragraph` form.
 - Accepts inputs of type:
 - * `string`
 - Defaults to `None`
- `set_event_tag_ids`
 - A list of Tag IDs or slugs to automatically apply to events created by this recipe.
 - Should be rendered with a `checkbox` form.
 - Choose from:
 - Accepts inputs of type:
 - * `string`
 - * `numeric`
 - Defaults to `[]`
- `set_event_content_items`
 - A list of Content Item IDs and Titles to automatically apply to events created by this Recipe.
 - Should be rendered with a `search` form.
 - Choose from:
 - Accepts inputs of type:
 - * `json`
 - Defaults to `[]`