pycrunchbase

Release 0.1.4

March 01, 2015

1 Overview **3**
1.1 pycrunchbase . . . 3

2 Installation **5**

3 Usage **7**

4 Reference **9**
4.1 pycrunchbase . . . 9

5 Contributing **11**
5.1 Bug reports . . . 11
5.2 Documentation improvements . . . 11
5.3 Feature requests and feedback . . . 11
5.4 Development . . . 11

6 Authors **13**

7 Changelog **15**
7.1 0.1.5 (2015-02-13) . . . 15
7.2 0.1.4 (2015-02-13) . . . 15
7.3 0.1.3 (2015-02-12) . . . 15
7.4 0.1.2 (2015-02-12) . . . 15
7.5 0.1.0 (2015-02-21) . . . 16

8 Indices and tables **17**

Python Module Index **19**

Contents:

CHAPTER 1

Overview

1.1 pycrunchbase

Python bindings to CrunchBase

1.1.1 Examples

```
# initialize the API using your API Key, will throw ValueError if missing
cb = CrunchBase(API_KEY)
# look up an organization by name
github = cb.organization('github')

# the response contains snippets of data regarding relationships
# that the organization has, an example is the funding_rounds
funding_rounds_summary = github.funding_rounds

# all relationships are paged, and only 8 is returned initially
# to get more data do this, it handles paging for you
# and returns a False-y value if there are no more pages
more_funding_rounds = cb.more(funding_rounds_summary)

# data in relations are just summaries, and you probably want more details
# For example funding_rounds returns 5 values: type, name, path
# created_at, updated_at.
# If you actually want to know who invested, you have to get to make
# more API calls

# first get the uuid of the round
round_uuid = funding_rounds_summary[0].uuid

# then use the CrunchBase API to make that call
round = cb.funding_round(round_uuid)

# again, investments is a relationship on a FundingRound,
# so we can get the first item in that relationship
an_investor = round.investments[0]  # a InvestorInvestmentPageItem
```

```
# and printing that gives us the name of the investor, and the amount
# invested in USD
print(str(an_investor))  # prints: Investor Name $100000
```

1.1.2 Installation

```
pip install pycrunchbase
```

1.1.3 Documentation

https://pycrunchbase.readthedocs.org/

1.1.4 Development

To run the all tests run:

```
tox
```

1.1.5 Goals

1. Support all (or almost all) of CrunchBase's API functionalities
2. Speedy updates when CrunchBase's API changes
3. 'Pythonic' bindings, user doesn't feel like we're requesting URLs

1.1.6 TODO

- Support other nodes (IPO, FundRaise)
- Coerce values in relationships page item to python types (datetime)
- Document our usage of returning None vs NonePageItem or NoneRelationship

1.1.7 License

MIT

CHAPTER 2

Installation

At the command line:

```
pip install pycrunchbase
```

Usage

To use pycrunchbase in a project:

```
import pycrunchbase
```

Instantiate your `CrunchBase` using your API Key:

```
cb = pycrunchbase.CrunchBase(API_KEY)
```

Get details about an organization:

```
github = cb.organization('github')
```

Get properties about the organization:

```
what_is_github = github.description
where_is_github = github.homepage_url
```

Get relationships (summarized version) about the organization:

```
github_team = github.current_team
who_started_github = github.founders
in_the_news = github.news
```

Get more relationships about the organization:

```
more_news = cb.more(in_the_news)
all_news_urls = [news.url for news in more_news]
```

If the relationship has more details to it, e.g. a Person in the current team, we need to do a bit more to grab those information:

```
a_founder = who_started_github[0]
cb.person(a_founder.cbid)
```

cbid is a special field on on an item in a relationship, this is the unique identifier of that node.

CHAPTER 4

Reference

4.1 pycrunchbase

class `pycrunchbase.`**`Acquisition`**(*data*)
Represents a Acquisition on CrunchBase API Docs: https://developer.crunchbase.com/docs

class `pycrunchbase.`**`FundingRound`**(*data*)
Represents a FundingRound on CrunchBase API Docs: https://developer.crunchbase.com/docs

class `pycrunchbase.`**`Organization`**(*data*)
Represents an Organization on CrunchBase API Docs: https://developer.crunchbase.com/docs

class `pycrunchbase.`**`Person`**(*data*)
Represents a Person on CrunchBase API Docs: https://developer.crunchbase.com/docs

class `pycrunchbase.`**`Product`**(*data*)
Represents a Product on CrunchBase API Docs: https://developer.crunchbase.com/docs

class `pycrunchbase.`**`Relationship`**(*name*, *data*)
A Relationhip represents relationship between a Node and interesting information regarding the Node. Relationships can be retrieved in two ways, via a call to a Node 1. /organization/name or a direct call to a relationship page 2. /organization/name/current_team We try to make this object easy to use by making this iterable, hiding the complexities of paging.

A call of type 1. will only get the summary of the relationships, to get the details we need to explicitly call methods on CrunchBase and retrieve them.

`__iter__`()
Allows callers to iterate through a relationship like this:

team_members = [member for member in company.current_team]

`get`(*i*)
Gets the i-th element of this relationship

Parameters **`i`** (*int*) – 0-based index of the element to retrieve

Returns if valid item exists at index i None if the index is too small or too large

Return type PageItem

class `pycrunchbase.`**`CrunchBase`**(*api_key=None*)
Class that manages talking to CrunchBase API

`acquisition`(*uuid*)
Get the details of a acquisition given a uuid.

Returns Acquisition or None

`funding_round`(*uuid*)
Get the details of a FundingRound given the uuid.

Returns FundingRound or None

`get_node`(*node_type*, *uuid*, *params=None*)
Get the details of a Node from CrunchBase. The node_type must match that of CrunchBase's, and the uuid is either the {uuid} or {permalink} as stated on their docs.

Returns containing the data describing this node with the keys uuid, type, properties, relationships. Or None if there's an error.

Return type dict

`more`(*relationship*)
Given a Relationship, tries to get more data using the next_page_url given in the response.

Returns None if there is no more data to get or if you have all the data Relationship with the new data

`organization`(*permalink*)
Get the details of a organization given a organization's permalink.

Returns Organization or None

`organizations`(*name*)
Search for a organization given a name, returns details of first match

Returns Organization or None

`person`(*permalink*)
Get the details of a person given a person's permalink

Returns Person or None

`product`(*permalink*)
Get the details of a product given a product permalink.

Returns Product or None

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

5.1 Bug reports

When reporting a bug please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.2 Documentation improvements

pycrunchbase could always use more documentation, whether as part of the official pycrunchbase docs, in docstrings, or even on the web in blog posts, articles, and such.

5.3 Feature requests and feedback

The best way to send feedback is to file an issue at https://github.com/ngzhian/pycrunchbase/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.4 Development

To set up *pycrunchbase* for local development:

1. Fork pycrunchbase on GitHub.
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/pycrunchbase.git
```

3. Create a branch for local development:

   ```
   git checkout -b name-of-your-bugfix-or-feature
   ```

   Now you can make your changes locally.

4. When you're done making changes, run all the checks, doc builder and spell checker with tox one command:

   ```
   tox
   ```

5. Commit your changes and push your branch to GitHub:

   ```
   git add .
   git commit -m "Your detailed description of your changes."
   git push origin name-of-your-bugfix-or-feature
   ```

6. Submit a pull request through the GitHub website.

5.4.1 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`) [1].
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

[1] If you don't have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.
It will be slower though ...

CHAPTER 6

Authors

- Ng Zhi An - http://github.com/ngzhian

CHAPTER 7

Changelog

7.1 0.1.5 (2015-02-13)

- Add a *cb_url* attribute for all PageItem, this url is a CrunchBase page (not the API) that holds more information for a particular PageItem Allows you to make calls like:

  ```
  company.funding_rounds[0].cb_url
  ```

  to get the url of the page for the first funding round of *company*.

- A new page item, InvestorInvestmentPageItem, that is useful for FundingRound info:

  ```
  round = cb.funding_round('round_uuid')
  an_investor = round.investments[0]  # a InvestorInvestmentPageItem
  print(str(an_investor))  # prints: Investor Name $100000
  ```

7.2 0.1.4 (2015-02-13)

- Relationship retrieval is 0-based now, 1-based just doesn't fit well with array
- Better *__str__* for *Node* and *Relationship*
- *Relationship.get(i)* if *i* is too large or small will return a NonePageItem singleton

7.3 0.1.3 (2015-02-12)

- Fix Relationship: wasn't using the right build method of PageItem
- Add test to checkk for the above
- remove unused reference to CrunchBase in Relationship

7.4 0.1.2 (2015-02-12)

- PageItem and it's subclasses to represent an item within a relationship of a Node
- Cleanup of where utility methods live (parse_date)

- More tests as always, overall 98.21% coverage

7.5 0.1.0 (2015-02-21)

- First release on PyPI.

CHAPTER 8

Indices and tables

- *genindex*
- *modindex*
- *search*

Python Module Index

p

pycrunchbase, 9

Index

Symbols

__iter__() (pycrunchbase.Relationship method), 9

A

Acquisition (class in pycrunchbase), 9
acquisition() (pycrunchbase.CrunchBase method), 9

C

CrunchBase (class in pycrunchbase), 9

F

funding_round() (pycrunchbase.CrunchBase method), 10
FundingRound (class in pycrunchbase), 9

G

get() (pycrunchbase.Relationship method), 9
get_node() (pycrunchbase.CrunchBase method), 10

M

more() (pycrunchbase.CrunchBase method), 10

O

Organization (class in pycrunchbase), 9
organization() (pycrunchbase.CrunchBase method), 10
organizations() (pycrunchbase.CrunchBase method), 10

P

Person (class in pycrunchbase), 9
person() (pycrunchbase.CrunchBase method), 10
Product (class in pycrunchbase), 9
product() (pycrunchbase.CrunchBase method), 10
pycrunchbase (module), 9

R

Relationship (class in pycrunchbase), 9