Contributors Documentation

Release 0.3.0

Daniel Roy Greenfeld

Jul 03, 2020

Contents

1 Contributors **3**
- 1.1 Usage . . . 3
- 1.2 Examples . . . 3
- 1.3 Credits . . . 4

2 Installation **5**
- 2.1 Stable release . . . 5
- 2.2 From sources . . . 5

3 Usage **7**

4 Contributing **9**
- 4.1 Types of Contributions . . . 9
- 4.2 Get Started! . . . 10
- 4.3 Pull Request Guidelines . . . 11
- 4.4 Tips . . . 11

5 Credits **13**
- 5.1 Development Leads . . . 13
- 5.2 Contributors . . . 13

6 History **15**
- 6.1 0.3.0 (2016-07-04) . . . 15
- 6.2 0.2.0 (2016-06-09) . . . 15
- 6.3 0.1.0 (2016-06-09) . . . 15

7 Indices and tables **17**

Contents:

CHAPTER 1

Contributors

A command-line script to get all the contributors for one or more GitHub projects.

- Free software: Apache Software License 2.0
- Documentation: https://contributors.readthedocs.io.

1.1 Usage

1. Add a *GITHUB_API_SECRET* to your environment variables
2. *pip install contributors*
3. Run the command:

```
$ contributors audreyr/cookiecutter,pydanny/cached-property
```

4. Wait while it processes
5. Open *output.rst* in the same directory you ran the script.

Warning: This may eat up a lot of GitHub bandwidth. If you are worred about that, do the following to limit the range of search:

```
$ contributors audreyr/cookiecutter,pydanny/cached-property -s 20160602 -u 20160610
```

1.2 Examples

- PyCon 2016 Cookiecutter sprint contributors

1.3 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

2.1 Stable release

To install Contributors, run this command in your terminal:

```
$ pip install contributors
```

This is the preferred method to install Contributors, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for Contributors can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/pydanny/contributors
```

Or download the tarball:

```
$ curl  -OL https://github.com/pydanny/contributors/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 3

Usage

To use Contributors in a project:

```
import contributors
```

CHAPTER 4

Contributing

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

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/pydanny/contributors/issues.

If you are 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.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/pydanny/contributors/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 :)

4.2 Get Started!

Ready to contribute? Here's how to set up *contributors* for local development.

1. Fork the *contributors* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/contributors.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv contributors
$ cd contributors/
$ python setup.py develop
```

4. Create a branch for local development:

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

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 contributors tests
$ python setup.py test or py.test
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. 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
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/pydanny/contributors/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_contributors
```

CHAPTER 5

Credits

5.1 Development Leads

- Daniel Roy Greenfeld <pydanny@gmail.com>
- Audrey Roy Greenfeld <audreyr@alum.mit.edu>

5.2 Contributors

- Saurabh Kumar / @theskumar
- Jon Banafato / @jonafato
- Leonard Ngonga

CHAPTER 6

History

- Add support for generating output in either markdown or rst with `-f` option.

6.1 0.3.0 (2016-07-04)

- Markdown support
- Fix input equality check when rate-limited
- Allow unauthenticated GitHub API calls
- Support configurable output filenames

6.2 0.2.0 (2016-06-09)

- Added pull requests and issues
- Made datetime not niave

6.3 0.1.0 (2016-06-09)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search