hashring

Release 1.5.1

Mar 05, 2018

Contents

Contents:

CHAPTER 1

Overview

1.1 hashring

build unknown BUILD FAILING coverage 71% code quality invalid
pypi v1.5.1 downloads no longer available

Implements consistent hashing in Python (using md5 as hashing function).

- Free software: BSD license

1.1.1 Installation

```
pip install hashring
```

1.1.2 Documentation

https://hashring.readthedocs.org/

1.1.3 Development

To run the all tests run:

```
tox
```

CHAPTER 2

Installation

At the command line:

```
pip install hashring
```

CHAPTER 3

Usage

To use hashring in a project:

```
import hashring
```

CHAPTER 4

Reference

4.1 hashring

CHAPTER 5

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

hashring could always use more documentation, whether as part of the official hashring 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/goller/hashring/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 *hashring* for local development:

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

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

- Chris Goller - https://github.com/goller/hashring

CHAPTER 7

Changelog

7.1 1.4.0 (2014-12-20)

- First release on PyPI.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index