sprockets.clients.cassandra

Release 0.0.0

May 12, 2015

Contents

Provides base functionality for asynchronously accessing/modifying data in a Cassandra cluster from within Tornado.

Althought the underlying library supports several connection options, this module currently only allows the hostname to be specified, and that via an environment variable called CASSANDRA_URI as specified in the docs.

CHAPTER 1

Documentation

https://sprocketsclientcassandra.readthedocs.org

CHAPTER 2

Contributing

This project follows the standard fork and pull request model of development. If you want to contribute changes, then fork the project and code away. To set up the environment:

- virtualenv env
- source env/bin/activate
- pip install -qr dev-requirements.txt

To test across supported platforms: * tox

To build the docs (in *build/sphinx/html*): * ./setup.py build_sphinx

CHAPTER 3

Version History

See https://github.com/sprockets/sprockets.clients.cassandra/blob/master/HISTORY.rst

CHAPTER 4

Documentation

4.1 How to Contribute

Do you want to contribute fixes or improvements?

> **AWesome!** *Thank you very much, and let's get started.*

This project is released under the permissive BSD license so you don't have to push changes back unless you want to. Since you are reading this, you have probably made the right decision.

4.1.1 Set up a development environment

The first thing that you need is a development environment so that you can run the test suite, update the documentation, and everything else that is involved in contributing. The easiest way to do that is to create a virtual environment for your endevours:

```
$ pyvenv env
```

Don't worry about writing code against previous versions of Python unless you you don't have a choice. That is why we run our tests through tox. If you don't have a choice, then install virtualenv to create the environment instead. The next step is to install the development tools that this project uses. These are listed in *dev-requirements.txt*:

```
$ env/bin/pip install -qr dev-requirements.txt
```

At this point, you will have everything that you need to develop at your disposal. *setup.py* is the swiss-army knife in your development tool chest. It provides the following commands:

./setup.py nosetests Run the test suite using nose and generate a nice coverage report.

./setup.py build_sphinx Generate the documentation using sphinx.

./setup.py flake8 Run flake8 over the code and report style violations.

If any of the preceding commands give you problems, then you will have to fix them **before** your pull request will be accepted.

4.1.2 Running Tests

The easiest (and quickest) way to run the test suite is to use the *nosetests* command. It will run the test suite against the currently installed python version and report not only the test result but the test coverage as well:

```
$ ./setup.py nosetests

running nosetests
running egg_info
writing dependency_links to sprockets.clients.cassandra.egg-info/dependency_links.txt
writing top-level names to sprockets.clients.cassandra.egg-info/top_level.txt
writing sprockets.clients.cassandra.egg-info/PKG-INFO
reading manifest file 'sprockets.clients.cassandra.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
warning: no previously-included files matching '__pycache__'...
warning: no previously-included files matching '*.swp' found ...
writing manifest file 'sprockets.clients.cassandra.egg-info/SOURCES.txt'
...

Name                       Stmts   Miss Branch BrMiss  Cover   Missing
----------------------------------------------------------------------
...
----------------------------------------------------------------------
TOTAL                         95      2     59      2    97%
----------------------------------------------------------------------
Ran 44 tests in 0.054s

OK
```

That’s the quick way to run tests. The slightly longer way is to run the **‘detox‘_** utility. It will run the test suite against all of the supported python versions in parallel. This is essentially what Travis-CI will do when you issue a pull request anyway:

```
$ env/bin/detox
py27 recreate: /.../sprockets.clients.cassandra/build/tox/py27
GLOB sdist-make: /.../sprockets.clients.cassandra/setup.py
py33 recreate: /.../sprockets.clients.cassandra/build/tox/py33
py34 recreate: /.../sprockets.clients.cassandra/build/tox/py34
py27 installdeps: -rtest-requirements.txt, mock
py33 installdeps: -rtest-requirements.txt
py34 installdeps: -rtest-requirements.txt
py27 inst: /.../sprockets.clients.cassandra/build/tox/dist/sprockets.clients.cassandra-0.0.0.zip
py27 runtests: PYTHONHASHSEED='2156646470'
py27 runtests: commands[0] | /../sprockets.clients.cassandra/build/tox/py27/bin/nosetests
py33 inst: /../sprockets.clients.cassandra/.build/tox/dist/sprockets.clients.cassandra-0.0.0.zip
py34 inst: /../sprockets.clients.cassandra/.build/tox/dist/sprockets.clients.cassandra-0.0.0.zip
py33 runtests: PYTHONHASHSEED='2156646470'
py33 runtests: commands[0] | /.../sprockets.clients.cassandra/build/tox/py33/bin/nosetests
py34 runtests: PYTHONHASHSEED='2156646470'
py34 runtests: commands[0] | /.../sprockets.clients.cassandra/build/tox/py34/bin/nosetests
_________________________________ summary _________________________________
  py27: commands succeeded
  py33: commands succeeded
  py34: commands succeeded
  congratulations :)
```

This is what you want to see. Now you can make your modifications and keep the tests passing.

4.1.3 Submitting a Pull Request

Once you have made your modifications, gotten all of the tests to pass, and added any necessary documentation, it is time to contribute back for posterity. You’ve probably already cloned this repository and created a new branch. If you haven't, then checkout what you have as a branch and roll back *master* to where you found it. Then push your repository up to github and issue a pull request. Describe your changes in the request, if Travis isn't too annoyed someone will review it, and eventually merge it back.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

CHAPTER 6

Release History

6.1 Next Release

- Implement greatness.