StoneRedis Documentation

Release 0.6.1

Pedro Ten

July 06, 2016

Contents:

StoneRedis

Redis client based on redis-py client with some added features

- Free software: ISC license
- Documentation: https://stoneredis.readthedocs.org.

1.1 Features

- Fully compatible with redis-py implementation
- Added some convenient features:
- Multi lpop: Pops multiple elements from a queue in an atomic way
- Multi rpush: Pushes multiple elements to a list. If bulk_size is set it will execute the pipeline every bulk_size elements. This operation will be atomic if transaction=True is passed
- Multi rpush limit: Pushes multiple elements to a list in an atomic way until it reaches certain size. Once limit is reached, the function will lpop the oldest elements. This operation runs in LUA, so is always atomic

1.2 Credits

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

Installation

At the command line:

```
$ easy_install stoneredis
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv stoneredis
$ pip install stoneredis
```

Usage

To use StoneRedis in a project:

```python
import stoneredis
```

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/stoneworksolutions/stoneredis/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

StoneRedis could always use more documentation, whether as part of the official StoneRedis 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/stoneworksolutions/stoneredis/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 *stoneredis* for local development.

1. Fork the *stoneredis* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/stoneredis.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 stoneredis
$ cd stoneredis/
$ 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 stoneredis tests
$ python setup.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/stoneworksolutions/stoneredis/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_stoneredis
```

Credits

5.1 Development Leadd

- Pedro Ten <pedroten@gmail.com>

5.2 Contributors

None yet. Why not be the first?

History

6.1 0.1.0 (2015-12-13)

- First release on PyPI.

Indices and tables

- genindex
- modindex
- search