
django-redisboard

Release 3.0.1

September 14, 2016

1	Overview	3
1.1	Overview	3
2	Installation	7
3	Usage	9
4	Reference	11
4.1	redisboard	11
5	Contributing	13
5.1	Bug reports	13
5.2	Documentation improvements	13
5.3	Feature requests and feedback	13
5.4	Development	13
6	Authors	15
7	Changelog	17
7.1	3.0.1 (2016-09-12)	17
7.2	3.0.0 (2015-12-17)	17
7.3	2.0.0 (2015-11-08)	17
7.4	1.2.2 (2015-10-11)	17
7.5	1.2.1 (2015-06-30)	17
7.6	1.2.0 (2015-02-21)	18
7.7	1.1.0 (2015-01-21)	18
7.8	1.0.0 (2014-12-10)	18
7.9	0.2.7 (?)	18
8	Indices and tables	19
	Python Module Index	21

Contents:

Overview

1.1 Overview

docs	
tests	
package	

Redis monitoring and inspection drop-in application using django admin.

- Free software: BSD license

1.1.1 Features

- Sever statistics in the admin changelist
- Key summary in the inspect view
- Value introspection with pagination for lists and sorted sets

1.1.2 Requirements

OS Any

Runtime Python 2.7, 3.4, 3.4 or PyPy

Services Redis 2.2 or later.

Packages Django>=1.8, py-redis>=2.10.0

1.1.3 Don't have a django project ?

Use the quick start script ! It will create and run a django project on 0.0.0.0:8000 with just the redisboard installed.

With curl:

```
curl -L https://raw.githubusercontent.com/ionelmc/django-redisboard/master/run_redisboard.py | tee run_redisboard.py
```

With wget:

```
wget --no-check-certificate https://raw.githubusercontent.com/ionelmc/django-redisboard/master/run_redisboard.py
```

Don't want to run on 0.0.0.0:8000 ? Run:

```
./run_redisboard.py ip:port
```

1.1.4 Installation guide

Install from pypi, with pip:

```
pip install django-redisboard
```

Or with setuptools:

```
easy_install django-redisboard
```

Add redisboard to INSTALLED_APPS:

```
INSTALLED_APPS += ("redisboard", )
```

After that you need to run:

```
manage.py syncdb
```

Or if you use south you can migrate this app:

```
manage.py migrate redisboard
```

Then you can add redis servers in the admin. You will see the stats in the changelist.

Redisboard has few css tweaks for the pages (they are optional). If you use staticfiles just run:

```
manage.py collectstatic
```

If you do not use django.contrib.staticfiles you must manually symlink the site-packages/redisboard/static/redisboard dir to <your media root>/redisboard.

1.1.5 Optional django settings

REDISBOARD_DETAIL_FILTERS

REDISBOARD_DETAIL_FILTERS - a list of regular expressions to match against the keys in the server details column. Eg, to only show uptime and list of active databases:

```
REDISBOARD_DETAIL_FILTERS = ['uptime.*', 'db.*']
```

To show all the details just use:

```
REDISBOARD_DETAIL_FILTERS = ['.*']
```

REDISBOARD_ITEMS_PER_PAGE

REDISBOARD_ITEMS_PER_PAGE - default 100. Used for paginating the items from a list or a sorted set.

REDISBOARD_SOCKET_TIMEOUT

REDISBOARD_SOCKET_TIMEOUT - default None. Socket operations time out after this many seconds.

REDISBOARD_SOCKET_CONNECT_TIMEOUT

REDISBOARD_SOCKET_CONNECT_TIMEOUT - default None. Socket connect operation times out after this many seconds.

REDISBOARD_SOCKET_KEEPAIVE

REDISBOARD_SOCKET_KEEPAIVE - default None. Enables or Disables socket keepalive.

REDISBOARD_SOCKET_KEEPAIVE_OPTIONS

REDISBOARD_SOCKET_KEEPAIVE_OPTIONS - default None. Additional options for socket keepalive.

1.1.6 Screenshots

Changelist:

Inspect:

Inspect key details:

1.1.7 Documentation

<https://django-redisboard.readthedocs.org/en/latest/>

1.1.8 Development

To run the all tests run:

tox

Installation

At the command line:

```
pip install django-redisboard
```

Usage

To use django-redisboard in a project:

```
import redisboard
```

Reference

4.1 redisboard

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

django-redisboard could always use more documentation, whether as part of the official django-redisboard 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/ionelmc/django-redisboard/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 code contributions are welcome :)

5.4 Development

To set up *django-redisboard* for local development:

1. Fork [django-redisboard](#) (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/django-redisboard.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 ...

Authors

- Ionel Cristian Mărie - <http://blog.ionelmc.ro>
- Areski Belaid - <http://areskibelaid.com/>
- ??? - <https://github.com/artscoop>
- Rick van Hattem - <http://w.wol.ph/>
- Benjamin Wohlwend - <https://github.com/piquadrat>
- Trbs - <https://github.com/trbs>
- John Lau - <https://github.com/jolks>
- ??? - <https://github.com/gabn88>

Changelog

7.1 3.0.1 (2016-09-12)

- Add support for Django 1.10. Contributed by Vincenzo Demasi in [#26](#).

7.2 3.0.0 (2015-12-17)

- Drop support for Django < 1.8
- Add support for Django 1.9. Contributed by gabn88 in [#25](#).

7.3 2.0.0 (2015-11-08)

- Fix error handling in couple places. Now pages don't return 500 errors if there's something bad going on with the redis connection.
- Remove key stats that came from `DEBUG OBJECT` (LRU, Address, Length etc). Now `OBJECT [REFCOUNT|ENCODING|IDLETIME]` is used instead. **BACKWARDS INCOMPATIBLE**

7.4 1.2.2 (2015-10-11)

- Exception handling for AWS ElastiCache Redis or any Redis that does not have `DEBUG OBJECT` command.
- Enabled Redis keys to be inspected despite not having details from `DEBUG OBJECT` command.

7.5 1.2.1 (2015-06-30)

- Fixed a bug on Python 3 (no `xrange`).
- Fixed some issues the `run_redisboard.py` bootstrapper had with `virtualenv`.

7.6 1.2.0 (2015-02-21)

- Add `REDISBOARD_SOCKET_TIMEOUT`, `REDISBOARD_SOCKET_CONNECT_TIMEOUT`, `REDISBOARD_SOCKET_KEEPALIVE` and `REDISBOARD_SOCKET_KEEPALIVE_OPTIONS` options.

7.7 1.1.0 (2015-01-21)

- Fix broken slowlog display.

7.8 1.0.0 (2014-12-10)

- Show slowlog and cpu usage and more memory stats (contributed by Rick van Hattem)
- Use pipelines to send commands for faster response (contributed by Rick van Hattem)
- Added Python 3.3 and 3.4 support.
- Added a test suite and other minor fixes.

7.9 0.2.7 (?)

- N/A.

Indices and tables

- `genindex`
- `modindex`
- `search`

r

redisboard, [11](#)

R

redisboard (module), [11](#)