
redis-throttled-queue

Release 1.0.0

Ionel Cristian Mărieș

Nov 15, 2022

CONTENTS

1	Overview	1
1.1	Installation	1
1.2	Documentation	1
1.3	Development	2
2	Installation	3
3	Usage	5
4	Reference	7
4.1	redis_throttled_queue	7
5	Contributing	9
5.1	Bug reports	9
5.2	Documentation improvements	9
5.3	Feature requests and feedback	9
5.4	Development	10
6	Authors	11
7	1.0.0 (2022-11-15)	13
8	0.6.0 (2022-07-06)	15
9	0.5.0 (2022-06-28)	17
10	0.4.4 (2022-05-09)	19
11	0.4.3 (2022-04-09)	21
12	0.4.2 (2022-04-02)	23
13	0.4.1 (2022-03-31)	25
14	0.4.0 (2022-03-31)	27
15	0.3.1 (2022-03-31)	29
16	0.3.0 (2022-03-31)	31
17	0.2.0 (2022-03-31)	33
18	0.1.0 (2022-03-30)	35

OVERVIEW

docs	
tests	
package	

Queue system with key-based throttling implemented over Redis.

- Free software: BSD 2-Clause License

1.1 Installation

```
pip install redis-throttled-queue
```

You can also install the in-development version with:

```
pip install https://github.com/ionelmc/python-redis-throttled-queue/archive/main.zip
```

1.2 Documentation

<https://python-redis-throttled-queue.readthedocs.io/>

1.3 Development

To run all the tests run:

<code>tox</code>

Note, to combine the coverage data from all the tox environments run:

Win-dows	<pre>set PYTEST_ADDOPTS=--cov-append tox</pre>
Other	<pre>PYTEST_ADDOPTS=--cov-append tox</pre>

INSTALLATION

At the command line:

```
pip install redis-throttled-queue
```

CHAPTER THREE

USAGE

To use redis-throttled-queue in a project:

```
import redis_throttled_queue
```


REFERENCE

4.1 `redis_throttled_queue`

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

redis-throttled-queue could always use more documentation, whether as part of the official redis-throttled-queue 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/python-redis-throttled-queue/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 *python-redis-throttled-queue* for local development:

1. Fork *python-redis-throttled-queue* (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:YOURGITHUBNAME/python-redis-throttled-queue.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 and docs builder 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*).
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 -- pytest -k test_myfeature
```

To run all the test environments in *parallel*:

```
tox -p auto
```

AUTHORS

- Ionel Cristian Mărieș - <http://blog.ionelmc.ro>

1 Changelog =====

1.0.0 (2022-11-15)

- Switched from eval scripts to redis functions. Minimum Redis server version becomes 7.0.
- Replaced unpack calls with direct indexing in the Lua functions.
- These changes improve the push operation by at least 6%.

0.6.0 (2022-07-06)

- Simplified `pop()` code to avoid the expensive scan operations. The `'...:names` key is now a sorted set and will be used as a template for the usage keys (`'...:usage:<window>'`).

0.5.0 (2022-06-28)

- Added support in a `AsyncThrottledQueue` class that only differs a bit from the regular `ThrottledQueue`:
 - `__len__` is removed, instead a awaitable `size()` method is available.
 - `__init__` doesn't validate version anymore, instead you can await on `validate_version()`.
 - `push()`, `pull()` and `cleanup()` are awaitable.
- Added a `validate_version` argument to `ThrottledQueue` (default: `True`).

0.4.4 (2022-05-09)

- Fixed missing usage key expiration when some queues are empty.

0.4.3 (2022-04-09)

- Fixed buggy counts when duplicate values are pushed. For now the highest priority will be used when two identical values would be pushed.

0.4.2 (2022-04-02)

- Refactored some duplicated code in the *pop* script.

0.4.1 (2022-03-31)

- Fixed bogus error in `cleanup()` when db is completely empty.

0.4.0 (2022-03-31)

- Added `last_activity` and `idle_seconds` attributes.
- Added a `cleanup()` method.

0.3.1 (2022-03-31)

- Renamed attributes (should be safe to mess with):
 - `_limit` becomes `limit`.
 - `_resolution` becomes `resolution`.

0.3.0 (2022-03-31)

- Allowed `pop(window)` using any window value (str/bytes/int recommended tho).

0.2.0 (2022-03-31)

- Fixed `__len__` (was returning a string).

0.1.0 (2022-03-30)

- First release on PyPI.

INDICES AND TABLES

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