
cheesecake_{*k*}*walitee*_{*i*}*index*Documentation

Release 0.1.1

Michael F Bryan

Jul 21, 2016

1	Cheesecake Kwalitee Index	3
1.1	Features	3
1.2	Credits	3
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
3	Usage	7
4	Architecture	9
4.1	Overall Design	9
5	cheesecake_kwalitee_index	11
5.1	cheesecake_kwalitee_index package	11
6	Contributing	13
6.1	Types of Contributions	13
6.2	Get Started!	14
6.3	Pull Request Guidelines	14
6.4	Tips	15
7	Credits	17
7.1	Development Lead	17
7.2	Contributors	17
8	Change Log	19
8.1	[Unreleased]	19
8.2	v0.1.1 (2016-19-20)	19
9	Indices and tables	21
	Python Module Index	23

Contents:

Cheesecake Kwalitee Index

A system that evaluates the *kwalitee* of all the packages on PyPI.

- Free software: MIT license
- Documentation: <https://cheesecake-kwalitee-index.readthedocs.io>.

1.1 Features

- TODO

1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

Installation

2.1 Stable release

To install `cheesecake_kwalitee_index`, run this command in your terminal:

```
$ pip install cheesecake_kwalitee_index
```

This is the preferred method to install `cheesecake_kwalitee_index`, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for `cheesecake_kwalitee_index` can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/Michael-F-Bryan/cheesecake_kwalitee_index
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/Michael-F-Bryan/cheesecake_kwalitee_index/tarball/  
↪master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

Usage

To use `cheesecake_kwalitee_index` in a project:

```
import cheesecake_kwalitee_index
```

Architecture

4.1 Overall Design

The system is designed in largely two parts, a front end and a back end. The back end will download and test all the packages on [PyPI](#), recording the results to a database. The front end then uses this data to compile an index that shows the *kwality* of each version of each package that has been tested.

Due to the sheer number of packages already uploaded to [PyPI](#) it will be necessary to distribute the testing among a large number of computers. Each testing node should test a package in a sandboxed environment (i.e. [Docker](#)), be almost completely stateless, and be designed in such a way that horizontal scaling is effortless.

4.1.1 Front End

The front end website should be fairly simple. It will most probably just be a [Django](#) site that reads from the central database and displays the data in an easily readable form.

4.1.2 Back End

The back-end will consist of a couple services working together, which may or may not be on the same box. These services are:

Relational Database Preferably Postgres.

Testing Daemon A program that will sit on a box, starting up new tests in their own Docker container when necessary and reporting test results to the database.

Message Queue To let the testing daemons know what package and version they should test next. Will most probably be implemented with a simple [Redis](#) queue

Master Node to keep an eye on the testing daemons and monitor their status, occasionally giving out new commands for it to execute (e.g. update, shut down, reload config etc.)

cheesecake_kwalitee_index

5.1 cheesecake_kwalitee_index package

5.1.1 Subpackages

cheesecake_kwalitee_index.kwalitee package

Submodules

cheesecake_kwalitee_index.kwalitee.evaluator module

The main suite of tests that are run to generate the cheesecake kwalitee index of a package.

```
class cheesecake_kwalitee_index.kwalitee.evaluator. Evaluator ( package,          ver-  
                                                         sion='*')
```

Bases: object

The Evaluator is in charge of downloading a package and evaluating it against the cheesecake kwalitee index.

clean_up ()

Remove the installed directory and do any other necessary clean up.

evaluate_score ()

Run the entire test suite and get the package's score.

get_version ()

Get `__version__`. If `__version__` exists then give 10 points, otherwise zero.

install_package ()

Try to install the package. If it installs, you get 10 points. Otherwise you get nothing.

lint_test ()

Pass the package through the linter.

`cheesecake_kwalitee_index.kwalitee.evaluator. download ( package, dest )`

Download a package to a specified location. Making sure to get a source distribution.

`cheesecake_kwalitee_index.kwalitee.evaluator. get_version_number (file_path,
 name)`

`cheesecake_kwalitee_index.kwalitee.evaluator. lint ( dest, name )`

Run Pylint and get the overall score of a package.

cheesecake_kwalitee_index.kwalitee.models module

All the data structures that are needed when generating a “kwalitee” score.

```
class cheesecake_kwalitee_index.kwalitee.models. Score ( value, total)  
    Bases: object  
    A generic score.  
    as_float ( )
```

Module contents

Seeing as the *cheesecake* package I was originally planning to use was designed for Python 2.x, I’m going to develop my own version of module to “Give a score to your Python package based on empirical ‘kwalitee’ factors”.

cheesecake_kwalitee_index.testing_daemon package

Module contents

The testing daemon for the *cheesecake_kwalitee_index*.

5.1.2 Submodules

5.1.3 cheesecake_kwalitee_index.utils module

Miscellaneous utility functions, classes, constants and decorators.

```
cheesecake_kwalitee_index.utils. get_logger ( name, log_file, log_level=None)  
    Get a logger object which is set up properly with the correct formatting, logfile, etc.
```

Parameters

- **name** (*str*) – The `__name__` of the module calling this function.
- **log_file** (*str*) – The filename of the file to log to.

Returns A logging.Logger object that can be used to log to a common file.

Return type logging.Logger

5.1.4 Module contents

A system that evaluates the kwalitee of all the packages on PyPI.

The *cheesecake_kwalitee_index* is a multi-part distributed system that checks out each version of each package on PyPI and runs the *cheesecake* ranking algorithm on the package in a sandboxed environment and reports the results to a central database server.

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:

6.1 Types of Contributions

6.1.1 Report Bugs

Report bugs at https://github.com/Michael-F-Bryan/cheesecake_kwalitee_index/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.

6.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

6.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

6.1.4 Write Documentation

`cheesecake_kwalitee_index` could always use more documentation, whether as part of the official `cheesecake_kwalitee_index` docs, in docstrings, or even on the web in blog posts, articles, and such.

6.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/Michael-F-Bryan/cheesecake_kwalitee_index/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 :)

6.2 Get Started!

Ready to contribute? Here's how to set up *cheesecake_kwalitee_index* for local development.

1. Fork the *cheesecake_kwalitee_index* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/cheesecake_kwalitee_index.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 cheesecake_kwalitee_index
$ cd cheesecake_kwalitee_index/
$ 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 cheesecake_kwalitee_index tests
$ python setup.py test or 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.

6.3 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/Michael-F-Bryan/cheesecake_kwalitee_index/pull_requests and make sure that the tests pass for all supported Python versions.

6.4 Tips

To run a subset of tests:

```
$ py.test tests.test_cheesecake_kwalitee_index
```

Credits

7.1 Development Lead

- Michael F Bryan <michaelfbryan@gmail.com>

7.2 Contributors

None yet. Why not be the first?

Change Log

A system that evaluates the kwalitee of all the packages on PyPI.

8.1 [Unreleased]

8.1.1 New Features

- [kwalitee] Started working on kwalitee (cheesecake clone) and made tests
- Started writing the Evaluator which will evaluate the quality of people's code
- [kwalitee] Added a test to see if **version** is defined

8.1.2 Documentation Changes

- Set up the changelog with auto-changelog

8.2 v0.1.1 (2016-19-20)

8.2.1 New Features

- Initial Commit

Indices and tables

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

C

`cheesecake_kwalitee_index`, [12](#)
`cheesecake_kwalitee_index.kwalitee`, [12](#)
`cheesecake_kwalitee_index.kwalitee.evaluator`,
 [11](#)
`cheesecake_kwalitee_index.kwalitee.models`,
 [12](#)
`cheesecake_kwalitee_index.testing_daemon`,
 [12](#)
`cheesecake_kwalitee_index.utils`, [12](#)

A

`as_float()` (`cheesecake_kwalitee_index.kwalitee.models.Score` method), 12

C

`cheesecake_kwalitee_index` (module), 12

`cheesecake_kwalitee_index.kwalitee` (module), 12

`cheesecake_kwalitee_index.kwalitee.evaluator` (module), 11

`cheesecake_kwalitee_index.kwalitee.models` (module), 12

`cheesecake_kwalitee_index.testing_daemon` (module), 12

`cheesecake_kwalitee_index.utils` (module), 12

`clean_up()` (`cheesecake_kwalitee_index.kwalitee.evaluator.Evaluator` method), 11

D

`download()` (in module `cheesecake_kwalitee_index.kwalitee.evaluator`), 11

E

`evaluate_score()` (`cheesecake_kwalitee_index.kwalitee.evaluator.Evaluator` method), 11

`Evaluator` (class in `cheesecake_kwalitee_index.kwalitee.evaluator`), 11

G

`get_logger()` (in module `cheesecake_kwalitee_index.utils`), 12

`get_version()` (`cheesecake_kwalitee_index.kwalitee.evaluator.Evaluator` method), 11

`get_version_number()` (in module `cheesecake_kwalitee_index.kwalitee.evaluator`), 11

I

`install_package()` (`cheesecake_kwalitee_index.kwalitee.evaluator.Evaluator` method), 11

`lint()` (in module `cheesecake_kwalitee_index.kwalitee.evaluator`), 11

`lint_test()` (`cheesecake_kwalitee_index.kwalitee.evaluator.Evaluator` method), 11

S

`Score` (class in `cheesecake_kwalitee_index.kwalitee.models`), 12