
Zertrin Python Toolbox Documentation

Release 0.2.0

Marc Gallet

Jul 04, 2020

Contents

1	Zertrin Python Toolbox	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	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	11
4.4	Tips	11
5	History	13
5.1	0.2.0 (2020-07-04)	13
5.2	0.1.1 (2019-08-29)	13
5.3	0.1.0 (2019-08-29)	13
5.4	0.0.10 (2018-05-30)	13
5.5	0.0.9 (2018-05-11)	14
5.6	0.0.8 (2018-05-11)	14
5.7	0.0.7 (2018-05-11)	14
5.8	0.0.6 (2018-04-17)	14
5.9	0.0.5 (2018-04-17)	14
5.10	0.0.4 (2017-06-27)	14
5.11	0.0.3 (2017-06-23)	14
5.12	0.0.2 (2017-06-22)	14
5.13	0.0.1 (2017-06-22)	15
6	Indices and tables	17

Contents:

A collection of useful functions by Zertrin

- Free software: MIT license
- Documentation: <https://zkpytb.readthedocs.io>.

1.1 Features

Module	Description
zkpytb.dicts	Helper functions for dictionaries.
zkpytb.json	Helpers related to json.
zkpytb.logging	Helper functions for logging.
zkpytb.pandas	Helper functions for Pandas.
zkpytb.priorityqueue	An implementation of PriorityQueue based on heapq.
zkpytb.utils	Miscellaneous helper functions.

1.2 Credits

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

2.1 Stable release

To install Zertrin Python Toolbox, run this command in your terminal:

```
$ pip install zkpytb
```

This is the preferred method to install Zertrin Python Toolbox, as it will always install the most recent 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 Zertrin Python Toolbox can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/zertrin/zkpytb
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/zertrin/zkpytb/tarball/master
```

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

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


CHAPTER 3

Usage

To use Zertrin Python Toolbox in a project:

```
import zkpytb
```

Usually, you only need a handful of functions from the toolbox:

```
from zkpytb.dicts import mergedicts  
from zkpytb.pandas import mad, percentile
```


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/zertrin/zkpytb/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” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

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

4.1.4 Write Documentation

Zertrin Python Toolbox could always use more documentation, whether as part of the official 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/zertrin/zkpytb/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 *zkpytb* for local development.

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

```
$ git clone git@github.com:your_name_here/zkpytb.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 zkpytb
$ cd zkpytb/
$ python setup.py develop
```

4. Create a branch for local development (please don't use *master*):

```
$ 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 zkpytb tests
$ 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 3.4, 3.5 and 3.6. Check https://travis-ci.org/zertrin/zkpytb/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_zkpytb_dicts
```


5.1 0.2.0 (2020-07-04)

- Update supported Python versions: drop Python 3.4 and added Python 3.8
- zkpytb.pandas: describe_numeric_1d(series): handle case where input series is empty

5.2 0.1.1 (2019-08-29)

- setup.py: add 'Programming Language :: Python :: 3.7' trove classifier

5.3 0.1.0 (2019-08-29)

- zkpytb.dicts: Add dict_value_map()
- zkpytb.pandas: add describe_numeric_1d(series)
- Add py 3.7 to tox.ini and .travis.yml

5.4 0.0.10 (2018-05-30)

- Add AutoDict and AutoOrderedDict classes in zkpytb.dicts
- zkpytb.dicts.hashdict and JsonEncoder: normalize path separator to ensure stable representation and hash for windows and linux.
- Fix tests test_compare_df_cols_*(*) to reach 100% coverage.

5.5 0.0.9 (2018-05-11)

- Add module `zkpytb.json` with a custom `JSONEncoder` class, and use it in `hashdict()`.

5.6 0.0.8 (2018-05-11)

- Add tests for `zkpytb.logging`

5.7 0.0.7 (2018-05-11)

- `zkpytb.dicts`: add `hashdict()` helper.

5.8 0.0.6 (2018-04-17)

- `zkpytb.pandas`: only try to import `statsmodels` when using `mad()`
- Minor changes missed while relasing previous version.

5.9 0.0.5 (2018-04-17)

- Add an implementation of `PriorityQueue` based on `heapqueue` in `zkpytb.priorityqueue`
- Add `mad(c)` (median absolute deviation) and `percentile(q)` functions in `zkpytb.pandas`
- Add code coverage and coveralls
- Add tests for `zkpytb.pandas`
- Fix `requirements_dev.txt` because `pandas>=0.21` is not compatible with `py34`

5.10 0.0.4 (2017-06-27)

- `zkpytb.utils`: add `hashstring()` and `get_git_hash()` helpers.
- Add tests for `zkpytb.dicts` and `zkpytb.utils` modules.

5.11 0.0.3 (2017-06-23)

- Add first version of `zkpytb.logging` module with function `setup_simple_console_and_file_logger()`.

5.12 0.0.2 (2017-06-22)

- Disable universal wheel.

5.13 0.0.1 (2017-06-22)

- First release on PyPI.

CHAPTER 6

Indices and tables

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