
tidytern

Release 0.0.0

Oct 07, 2019

Contents

1	tidytern	1
1.1	Installation	1
1.2	Documentation	1
1.3	Development	2
1.4	TODOs	2
1.5	Credits	2
2	Installation	3
3	Usage	5
4	Reference	7
4.1	tidytern	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	Changelog	13
7.1	0.0.0 (2019-09-27)	13
8	Indices and tables	15

CHAPTER 1

tidytern

docs	
tests	
package	

Disciplined versioned schema migration of PostgreSQL databases

1.1 Installation

```
pip install tidytern
```

You can also install the in-development version with:

```
pip install https://github.com/agravier/tidytern/archive/master.zip
```

1.2 Documentation

<https://tidytern.readthedocs.io/>

1.3 Development

Python versions (both for development and for tox tests) and are most easily managed with [Pyenv](#) and its virtualenv plugin.

To install the latest supported Python versions with pyenv and list them in the local .python-version file for use by tox:

```
pyenv install 3.7.4 && pyenv install 3.8.0b4
```

1.3.1 Running tests

The test runner is `pytest`. The tool used to automate running various types of tests under different environments is `tox`.

Install tox in the global Python 3 environment. It may be installed using your package manager or with pip:

```
pip3 install tox
```

To run the all tests, ensure that you are not within a virtualenv and run:

```
tox
```

To combine the coverage data from all the tox environments run:

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

Two requirements files are used to describe the development setup:

- The requirements.txt file describes a working development environment with all pinned dependencies.
- The requirements-base.txt file contains the direct unpinned dependencies only.

1.4 TODOs

- Change ‘3.8-dev’ to ‘3.8’ once the final release of Python 3.8 has landed

1.5 Credits

The initial project was setup with the help of [ionelmc’s cookiecutter template](#).

CHAPTER 2

Installation

At the command line:

```
pip install tidytern
```


CHAPTER 3

Usage

To use tidytern in a project:

```
import tidytern
```


CHAPTER 4

Reference

4.1 tidytern

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

tidytern could always use more documentation, whether as part of the official tidytern 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/agravier/tidytern/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 *tidytern* for local development:

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

```
git clone git@github.com:agravier/tidytern.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 -- pytest -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 ...

CHAPTER 6

Authors

- Alexandre Gravier - <http://agravier.com>

CHAPTER 7

Changelog

7.1 0.0.0 (2019-09-27)

- First release on PyPI.

CHAPTER 8

Indices and tables

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