
whatsonpypi Documentation

Release 0.2.7

Visesh Prasad

Feb 13, 2019

Contents:

1	whatsonpypi	1
1.1	Installation	1
1.2	Requirements	1
1.3	Features	1
1.4	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
4.5	Deploying	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.2.7 (2019-02-12)	15
6.2	0.2.6 (2019-02-06)	15
6.3	0.2.5 (2019-02-05)	15
6.4	0.2.4 (2019-01-29)	15
6.5	0.2.3 (2019-01-26)	15
6.6	0.2.2 (2019-01-23)	16
6.7	0.2.1 (2019-01-23)	16
6.8	0.2.0 (2019-01-22)	16
6.9	0.1.2 (2019-01-20)	16
6.10	0.1.1 (2019-01-02)	16
7	Indices and tables	17

CLI tool to get package info from PyPI and/or manipulate requirements.

- GitHub: <https://github.com/viseshrp/whatsonpypi>
- PyPI: <https://pypi.python.org/pypi/whatsonpypi>
- Free software: MIT license
- Documentation: <https://whatsonpypi.readthedocs.io>.

1.1 Installation

```
pip install -U whatsonpypi
```

1.2 Requirements

1. Python 2.7+

1.3 Features

- Find information on a package on PyPI

Examples:

```
$ whatsonpypi django
NAME
  Django
LATEST VERSION
  2.1.5
```

(continues on next page)

(continued from previous page)

```
SUMMARY
  A high-level Python Web framework that encourages rapid development,
  ↪and clean, pragmatic design.
PACKAGE URL
  https://pypi.org/project/Django/
AUTHOR
  Django Software Foundation
LATEST RELEASES
  2.2a1, 2.1rc1, 2.1b1, 2.1a1, 2.1.5
```

- For more information..

Examples:

```
$ whatsonpypi django --more
...
```

- Version specific information..

Examples:

```
$ whatsonpypi django==2.1.4 --more
...
```

- Launch documentation URL of project in a browser tab

Examples:

```
$ whatsonpypi django --docs
```

- Add packages to your requirements files.

Examples:

```
$ whatsonpypi django --add
```

By default, it searches for files with names matching `requirements*.txt` in the current working directory and adds the dependency to the end of the file.

You can change the filename pattern to search for. The pattern may contain simple shell-style wild-cards.

```
$ whatsonpypi django --add --req-pattern "*.txt"
```

If there's more than one file, you will see a prompt allowing you to select the files that should be modified.

If you want the dependency to be added to a specific line, mention a comment `#wopp` on its own line, which will be replaced with the dependency:

Example:

Do this in your requirements.txt:

```
# Django
django==2.1.5
# testing
pytest==4.1.1
#wopp
```

Then running this:

```
$ whatsonpypi pytest-runner --add
```

will produce this:

```
# Django
django==2.1.5
# testing
pytest==4.1.1
pytest-runner==4.2
```

Existing dependencies will be replaced with newer versions. Dependency version by default is the latest unless specified explicitly like:

```
$ whatsonpypi pytest-runner==4.1 --add
```

Optionally, directory to search for requirement files can be specified with `--req-dir`. Both absolute and relative paths are allowed. Must be a directory.

```
$ whatsonpypi pytest-runner==4.1 --add --req-dir /Users/Me/Documents/
↳GitHub/project/requirements
```

Default value (if not provided) is the directory where the command is run (cwd).

Also, optionally, you can specify comments to add before a dependency. Note that the comment will not be added if the dependency already exists in the file.

For example, running this:

```
$ whatsonpypi pytest-runner --add --comment 'testing'
```

will add this:

```
# testing
pytest-runner==4.2
```

See all options with:

```
$ whatsonpypi --help
```

1.4 Credits

- Cookiecutter and the [audreyr/cookiecutter-pypackage](#) project template for getting me started.

2.1 Stable release

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

```
$ pip install whatsonpypi
```

This is the preferred method to install `whatsonpypi`, 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 `whatsonpypi` can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/viseshrp/whatsonpypi
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/viseshrp/whatsonpypi/tarball/master
```

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

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


CHAPTER 3

Usage

To use whatsonypyi in a project:

```
import whatsonypyi
```


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/viseshrp/whatsonpypi/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

whatsonypyi could always use more documentation, whether as part of the official whatsonypyi 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/viseshrp/whatsonypyi/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 *whatsonypyi* for local development.

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

```
$ git clone git@github.com:your_name_here/whatsonypyi.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 whatsonypyi
$ cd whatsonypyi/
$ 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 whatsonypyi 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.

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 2.7+, and for PyPy. Check https://travis-ci.org/viseshrp/whatsonpypi/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_whatsonpypi
```

4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

5.1 Development Lead

- Visesh Prasad <viseshrprasad@gmail.com>

5.2 Contributors

None yet. Why not be the first?

6.1 0.2.7 (2019-02-12)

- fix for py2 ImportError

6.2 0.2.6 (2019-02-06)

- fix absence of new lines

6.3 0.2.5 (2019-02-05)

- fix requirements file format check

6.4 0.2.4 (2019-01-29)

- add `-comment` to allow mentioning comments with `-add`

6.5 0.2.3 (2019-01-26)

- add `-req-pattern` to allow specifying search pattern for requirements files
- raise Error when no requirements files are found

6.6 0.2.2 (2019-01-23)

- Fix empty dependencies display

6.7 0.2.1 (2019-01-23)

- Small fixes

6.8 0.2.0 (2019-01-22)

- Add `-a/--add` to enable adding packages to requirement files.
- Added `-d/--docs` to launch docs URL in browser
- Allow version specific querying

6.9 0.1.2 (2019-01-20)

- Display more package info
- added `--more/-m` for even more info

6.10 0.1.1 (2019-01-02)

- First release on PyPI.

CHAPTER 7

Indices and tables

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