

---

# Imports Documentation

*Release 0.1.4*

**João Paulo Canário**

**Oct 13, 2020**



---

## Contents:

---

<b>1 Imports</b>	<b>1</b>
1.1 Features . . . . .	1
1.2 Credits . . . . .	1
<b>2 Installation</b>	<b>3</b>
2.1 Stable release . . . . .	3
2.2 From sources . . . . .	3
<b>3 Usage</b>	<b>5</b>
<b>4 imports</b>	<b>7</b>
4.1 imports package . . . . .	7
<b>5 Contributing</b>	<b>9</b>
5.1 Types of Contributions . . . . .	9
5.2 Get Started! . . . . .	10
5.3 Pull Request Guidelines . . . . .	11
5.4 Tips . . . . .	11
5.5 Deploying . . . . .	11
<b>6 Credits</b>	<b>13</b>
6.1 Development Lead . . . . .	13
6.2 Contributors . . . . .	13
<b>7 History</b>	<b>15</b>
7.1 0.1.0 (2018-03-20) . . . . .	15
7.2 0.1.2 (2018-03-22) . . . . .	15
7.3 0.1.4 (2018-03-23) . . . . .	15
<b>8 Indices and tables</b>	<b>17</b>
<b>Python Module Index</b>	<b>19</b>
<b>Index</b>	<b>21</b>



A helper tool to identify unused installed packages

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

## 1.1 Features

- List all installed packages added on the project requirements that are not used

## 1.2 Credits

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



### 2.1 Stable release

To install Imports, run this command in your terminal:

```
$ pip install imports
```

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

You can either clone the public repository:

```
$ git clone git://github.com/joaopcanario/imports
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/joaopcanario/imports/tarball/master
```

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

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





## CHAPTER 3

---

### Usage

---

To use Imports in a project:

```
from imports import imports as imp  
imp.check('path_to_project_folder')
```



## 4.1 imports package

### 4.1.1 Submodules

### 4.1.2 imports.cli module

### 4.1.3 imports.imports module

```
imports.imports.check(path_dir, requirements_name='requirements.txt')
```

Look for unused packages listed on project requirements

### 4.1.4 Module contents

Top-level package for Imports.



Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

## 5.1 Types of Contributions

### 5.1.1 Report Bugs

Report bugs at <https://github.com/joaopcanario/imports/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.

### 5.1.2 Fix Bugs

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

### 5.1.3 Implement Features

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

### 5.1.4 Write Documentation

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

### 5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/joaopcanario/imports/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 :)

## 5.2 Get Started!

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

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

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

## 5.3 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, 3.4, 3.5 and 3.6, and for PyPy. Check [https://travis-ci.org/joaopcanario/imports/pull\\_requests](https://travis-ci.org/joaopcanario/imports/pull_requests) and make sure that the tests pass for all supported Python versions.

## 5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_imports
```

## 5.5 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.





#### 6.1 Development Lead

- João Paulo Canário <jopacanario@gmail.com>

#### 6.2 Contributors

- Augusto Goulart <augusto@goulart.me>



### 7.1 0.1.0 (2018-03-20)

- First release on PyPI.

### 7.2 0.1.2 (2018-03-22)

- Fix minor issues.
- Update documentation
- Minor code improvements

### 7.3 0.1.4 (2018-03-23)

- Improve tests
- Fix minor issues



## CHAPTER 8

---

### Indices and tables

---

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



### i

`imports`, [7](#)

`imports.imports`, [7](#)





## C

`check()` (*in module `imports.imports`*), 7

## I

`imports(module)`, 7

`imports.imports(module)`, 7