
fullcoverage Documentation

Release 0.1.0

Louis Paternault

Oct 07, 2023

CONTENTS

1	What's new?	3
2	Download and install	5
3	How-to	7
4	Bugs and Limitations	9
5	FAQ	11

In case this is not clear enough: *this module is a joke !* Do not use it. It is not dangerous; it is just useless.

Nowadays, every trendy project has a collection of shiny little badges. They look great, but they can be hard to get. This module is here to help you to get the 100 % test coverage badge, without any effort.

- [*What's new?*](#)
- [*Download and install*](#)
- [*How-to*](#)
- [*Bugs and Limitations*](#)
- [*FAQ*](#)

WHAT'S NEW?

See [changelog](#).

DOWNLOAD AND INSTALL

- From sources:
 - Download: <https://pypi.python.org/pypi/fullcoverage>
 - Install (in a *virtualenv*, if you do not want to mess with your distribution installation system):

```
python3 setup.py install
```

- From pip:

```
pip install fullcoverage
```

- Quick and dirty Debian (and Ubuntu?) package

This requires [stdeb](#) to be installed:

```
python3 setup.py --command-packages=stdeb.command bdist_deb  
sudo dpkg -i deb_dist/fullcoverage-<VERSION>_all.deb
```


HOW-TO

Let's say you want to get you module `foobar` thoroughly covered.

1. Install *fullcoverage*:

```
pip install fullcoverage
```

2. Enable this plugin. That is, in the `.coveragerc` file of `foobar`, have the lines:

```
[run]
plugins = fullcoverage.plugin
```

3. Optionnaly, if you want to restrict the test report to your module, you can add, in the same `.coveragerc` file:

```
[fullcoverage.plugin]
source = foobar
```

4. If you already have tests that import all of `foobar` submodules, that's it. Otherwise, somewhere in your tests, have the following lines import everything in your module:

```
import foobar
import fullcoverage

fullcoverage.import_all(foobar)
```

5. That's it! You can now run your tests:

```
coverage run --source foobar -m unittest
coverage report --fail-under=100
```


BUGS AND LIMITATIONS

- This does not work with `doctest`. Please feel free to [send a patch](#).
- Even though tests cover 100 % of your module, it may still contain bugs.

FAQ

- Isn't it dangerous? Are you running every line of a module to get 100 % test coverage? What if you run a line `shutil.rmtree('/')` ?

This plugins does not run random line of codes. It imports every plugin that is to be covered, but it cheats coverage into thinking they were thoroughly tested, without testing anything. If your module is safe to import, it is safe to use `fullcoverage`. If your module is not safe to import, fix your module.