
pytest-requests-futures

Release 0.9.0

Konstantinos Lampridis

Jul 06, 2022

CONTENTS:

1	Features	3
1.1	Development	3
2	Prerequisites	5
3	Quickstart	7
4	License	9
5	License	11
5.1	Introduction	11
5.2	Why this Package?	11
5.3	Usage	11
5.4	pytest_requests_futures	12
6	Indices and tables	13
	Python Module Index	15
	Index	17

MOCKING REQUESTS FUTURES

Pytest Plugin to Mock Requests Futures

Code: <https://github.com/boromir674/pytest-requests-futures>

Docs: <https://pytest-requests-futures.readthedocs.io/en/master/>

PyPI: <https://pypi.org/project/pytest-requests-futures/>

CI: <https://github.com/boromir674/pytest-requests-futures/actions/>

FEATURES

1. **pytest_requests_futures** *python package*
 - a. TODO Document a **Great Feature**
 - b. TODO Document another **Nice Feature**
2. Tested against multiple *platforms* and *python* versions

1.1 Development

Here are some useful notes related to doing development on this project.

1. **Test Suite**, using `pytest`, located in *tests* dir
2. **Parallel Execution** of Unit Tests, on multiple cpu's
3. **Documentation Pages**, hosted on *readthedocs* server, located in *docs* dir
4. **Automation**, using `tox`, driven by single *tox.ini* file
 - a. **Code Coverage** measuring
 - b. **Build Command**, using the `build` python package
 - c. **Pypi Deploy Command**, supporting upload to both `pypi.org` and `test.pypi.org` servers
 - d. **Type Check Command**, using `mypy`
 - e. **Lint Check** and *Apply* commands, using `isort` and `black`
5. **CI Pipeline**, running on `Github Actions`, defined in *.github/*
 - a. **Job Matrix**, spanning different *platform's* and *python version's*
 1. Platforms: *ubuntu-latest*, *macos-latest*
 2. Python Interpreters: *3.6*, *3.7*, *3.8*, *3.9*, *3.10*
 - b. **Parallel Job** execution, generated from the *matrix*, that runs the *Test Suite*

PREREQUISITES

You need to have *Python* installed.

QUICKSTART

Using *pip* is the approved way for installing *pytest_requests_futures*.

```
python3 -m pip install pytest_requests_futures
```

TODO Document a use case

LICENSE

- [GNU Affero General Public License v3.0](#)

- Free software: GNU Affero General Public License v3.0

5.1 Introduction

This is **Mocking Requests Futures**, a *Python Package* desinged to ...

Goal of this project is to TODO Document
Additionally, TODO Document

This documentation aims to help people understand what are the package's features and to demonstrate how to leverage them for their use cases.
It also presents the overall package design.

5.2 Why this Package?

So, why would one opt for this Package?

It is **easy** to *install* (using *pip*) and intuitive to *use*.

Mocking Requests Futures features TODO Document

Well-tested against multiple Python Interpreter versions (3.6 - 3.10), tested on both *Linux* (Ubuntu) and *Darwin* (Macos) platforms.

Tests trigger automatically on **CI**. The package's releases follow **Semantic Versioning**.

5.3 Usage

5.3.1 Installation

pytest_requests_futures is available on PyPI hence you can use *pip* to install it.

It is recommended to perform the installation in an isolated *python virtual environment* (env). You can create and activate an *env* using any tool of your preference (ie *virtualenv*, *venv*, *pyenv*).

Assuming you have ‘activated’ a *python virtual environment*:

```
python -m pip install pytest-requests-futures
```

5.3.2 Simple Use Case

Common Use Case for the `pytest_requests_futures` is to TODO Document

TODO Document

5.4 `pytest_requests_futures`

5.4.1 `pytest_requests_futures` package

Module contents

`pytest_requests_futures.future_mock()`

Build a mock of an instance of a Future.

Creates a mock object, that behaves similar to a Future instance, given an http status code, which the client desires the mock to emulate.

The mock object is similar to the output of the `MockFuturesSession.get` method and has the single “result” ‘method’ that returns an object with the “status_code” property.

Parameters `status_code` (*int*) – the http status code that the client desires the mock to return

Returns an instance of a Futere Mock

Return type FutureMock

`pytest_requests_futures.future_session_mock(future_mock)`

Build a mock of an instance of a FutureSession.

Create a mock object that partially emulates the behaviour of a `MockFuturesSession` instance and “exposes” a mocked ‘get’ method.

`pytest_requests_futures.future_session_mock_from_boolean(future_session_mock)`

Mock the `FuturesSession` class.

Get a mock of the `FuturesSession` class.

Parameters `found` (*bool*) – whether to emulate the mock returning 200 or not as an http “status code”

INDICES AND TABLES

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

PYTHON MODULE INDEX

p

`pytest_requests_futures`, [12](#)

INDEX

F

`future_mock()` (*in module `pytest_requests_futures`*), [12](#)
`future_session_mock()` (*in module `pytest_requests_futures`*), [12](#)
`future_session_mock_from_boolean()` (*in module `pytest_requests_futures`*), [12](#)

M

`module`
 `pytest_requests_futures`, [12](#)

P

`pytest_requests_futures`
 `module`, [12](#)