**PyTest**

* **Naming Conventions:**
* **Running pytest from command line**
* **Running selected test from command line**
* **Running selected test method using matching keyword**
* **Failing and skipping tests using annotations**
* **Fixtures and hooks**

## Naming Conventions:

Any pytest file should start with keyword called **test\_**

**Ex: test\_demo.py**

## Running pytest from command line

* Go to file locations: C:\Users\nursh\PycharmProjects\PySelenium\pytestDemo>
* Then enter **py.test**
* It will execute all test under pytestDemo file
* To Run with verbose: **py.test -v**
* To run with verbose and details: **py.test -v -s**

## Running single test file from command line

Command: py.test test\_filename.py

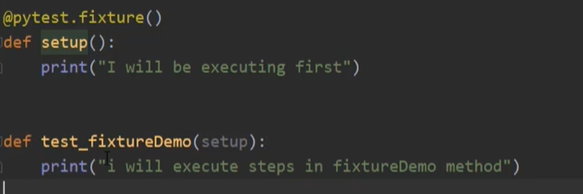


Fixtures

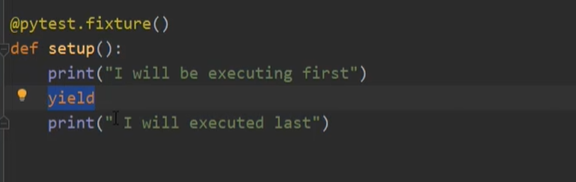
Fixtures are functions, which will run before and/or after each test function to which it is applied. Fixtures are used to feed some data to the tests such as database connections, URLs to test and some sort of input data.

A function is marked as a fixture by −

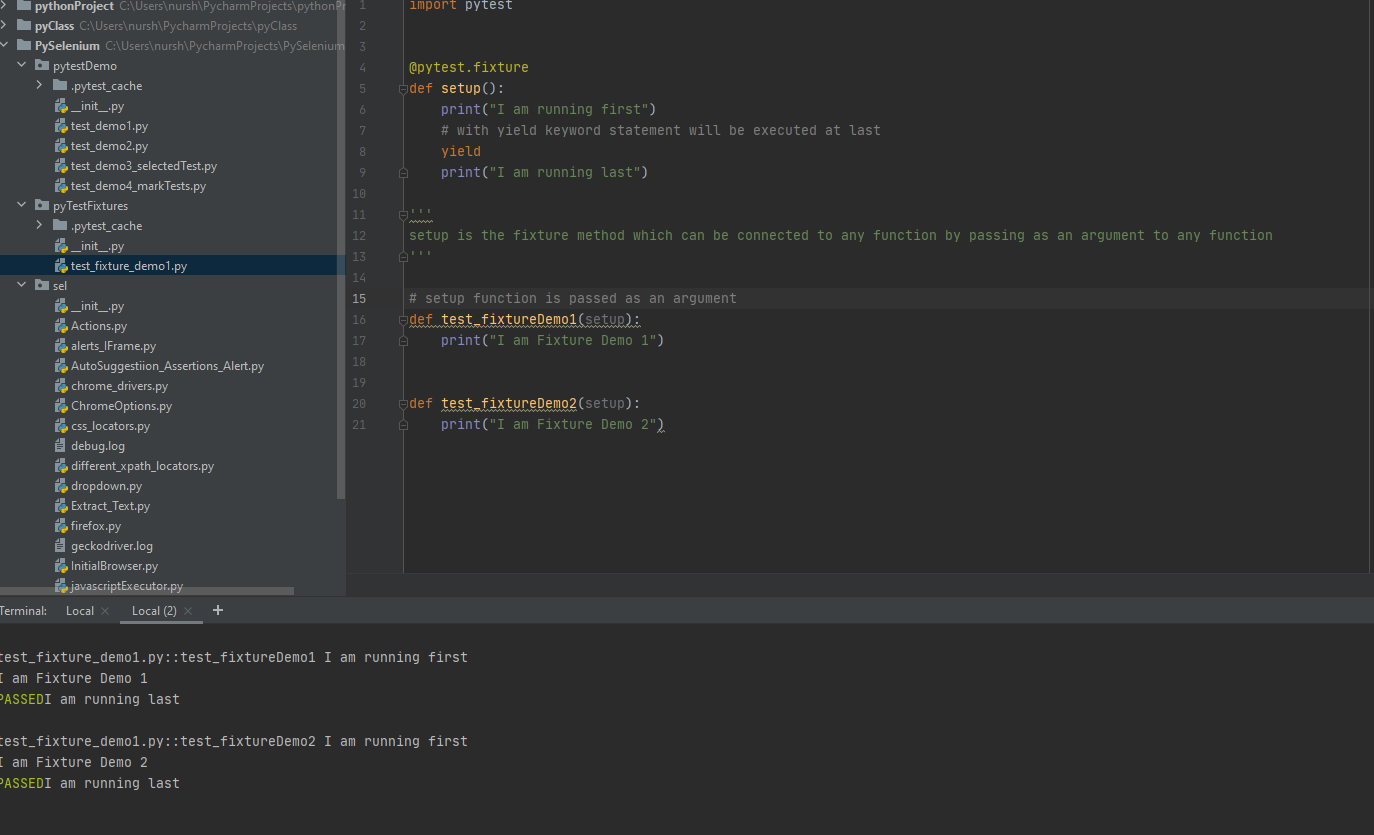
**@pytest.fixture**



Setup is the fixture method and passed as an argument to test\_fixtureDemo function. When test\_fixtureDemo is executed it will trigger setup function first.

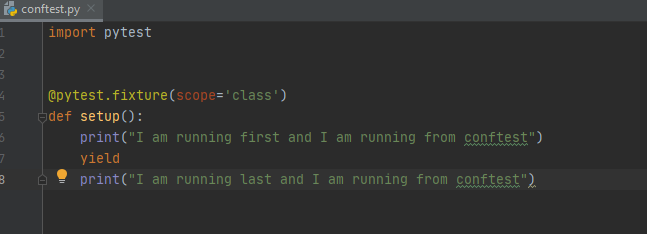


If user wants to run any function at the end of all test user can do it using same fixture with yield keyword.

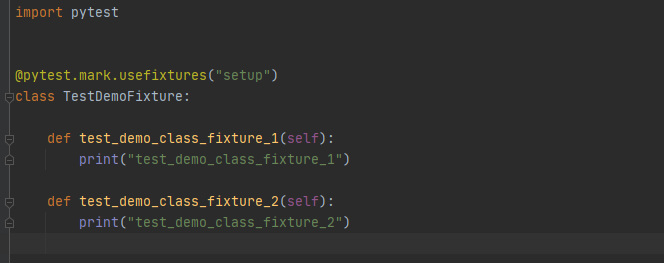


**To set up fixture for every test, it can be done when all tests wrapped under a class.**

**To do that define a file called “conftest.py”**

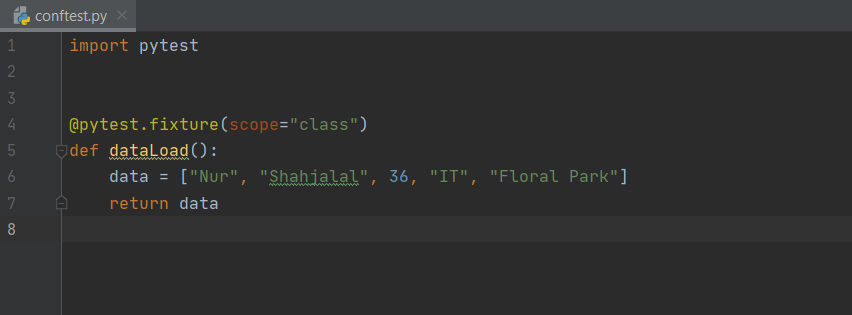
****

**Actual test :**

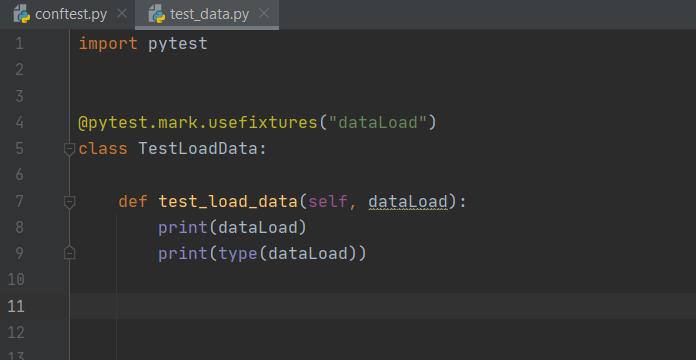
****

# Data Driven Test – Parameterizing Test

Define a conftest.py file



Actual test:



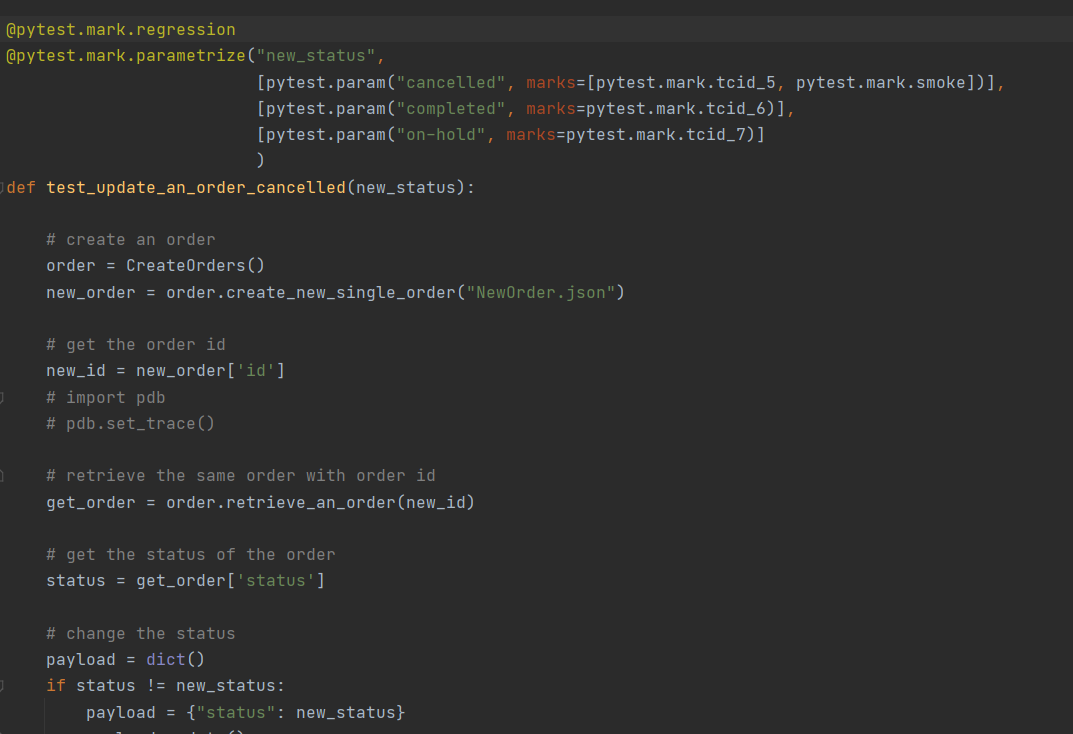
# PyTest Command line arguments for browser choice

<https://docs.pytest.org/en/2.0.3/example/simple.html>

# **Parameterize Tests:**

If a single test has multiple scenario and can be run just changing a parameter with different test id.

Suppose , following test is to validate the order status like ( completed, on-hold, cancelled)



This test can be run all together:

* Pytest -m regression

A single test can be run as well

Pytest -m tcid\_5 >> then the test is going to validate “cancelled” status

# PyTest HTML Report Generation

Pip install pytest-html

To generate report :

Py.test - -html=report.html

Some useful command

* Where python
* Python –version’
* Pip list



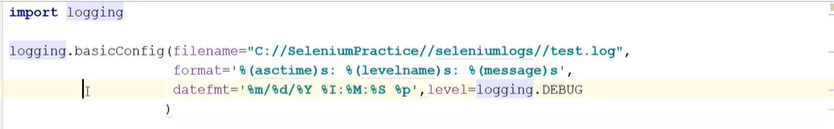


<https://pypi.org/project/testproject-python-sdk/>

# Logging

To capture log file in html report :

**py.test --html=new\_report.html --self-contained-html -s --capture=tee-sys**

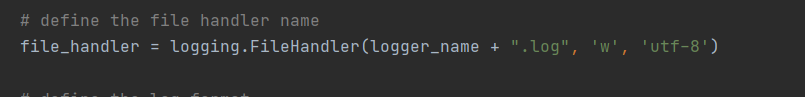


**Problem:**

Logging **charmap** error,

Solutions:

Add “utf-8” to filehandler



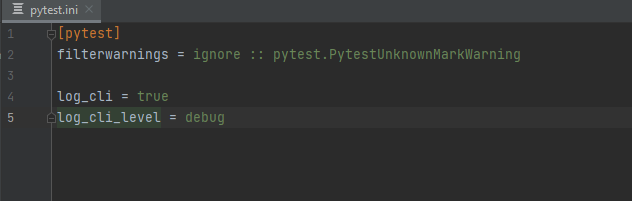
**Pytest Unknown Marking warning:**

If any test has no registered marking ( **@pytest.mark.tcid1** ), pytest will complain about unknown markings, to avoid this :

# registered the mark

Or

# set up an pytest.ini file under the project where the file has the command to ignore the warning



**Module Not Found Error**

It can happen if any module missed The “\_\_init\_\_.py” file.

All the modules “from Import” and “to import” module must have a

Empy “\_\_init\_\_.py” file

**Different status code:**

200 = ok

201 = post ok

400 = bad request

401 = unauthorized

**Type Of Authentications:**

* Basic authentication
  + HTTPBasicAuth ( username and password)
* Digest authentication
  + HTTPDigestAuth**(**'user'**,** 'pass'**)**
* Oauth 1 Authentication
  + OAuth1**(**'YOUR\_APP\_KEY'**,** 'YOUR\_APP\_SECRET'**,**

'USER\_OAUTH\_TOKEN'**,** 'USER\_OAUTH\_TOKEN\_SECRET'**)**

* OAuth 2 Authentication
* Other Authentication
* New Forms of Authentication