

# Testing using pytest in Python

## Overview

In this tutorial, we will create some tests in Python using [pytest](#).



### Please note

There is a [more up-to-date tutorial using pytest](#) providing additional capabilities. Please check it instead.

## Requirements

- Install the Python "pip"
- Install "pytest"

```
pip install -U pytest
```

## Description

Below are two Python scripts implementing tests with subtle differences.

### test\_sample.py

```
def func(x):  
    return x + 1  
def test_answer():  
    assert func(3) == 5
```

### test\_class.py

```
class TestClass:  
    def test_one(self):  
        x = "this"  
        assert 'h' in x  
    def test_two(self):  
        x = "hello"  
        assert hasattr(x, 'check')
```

After running the tests and generating the JUnit XML report (e.g., [py-results1.xml](#)), it can be imported to Xray (by using either the REST API or the **Import Execution Results** action within the Test Execution).

```
pytest --junitxml=py-results1.xml
```

JUnit's Test Case is mapped to a Generic Test in Jira, and the **Generic Test Definition** field contains "pytest" concatenated with the name of the method that implements the test case.

The Execution Details of the Generic Test contains information about the Test Suite, which in this case corresponds to "pytest".

## Execution Details

### Test Description

None

### Test Details

Test Type: Generic  
Definition: `test_class.TestClass.test_one`

### Results

| Context          | Error Message | Duration    | Status        |
|------------------|---------------|-------------|---------------|
| TestSuite pytest | -             | 1 millisecc | <b>PASSED</b> |

## References

- <http://docs.python-guide.org/en/latest/writing/tests/>
- <http://doc.pytest.org/en/latest/usage.html#creating-junitxml-format-files>