



TESTY AUTOMATYCZNE W JĘZYKU PYTHON

Rafał Błaczkowski
Deep Learning Software Engineer

Internet of Things Group

PYTEST

- Darmowy framework testowy
- Język programowania: Python
- Bogata biblioteka gotowych funkcjonalności ułatwiających testowanie
- Dobrze udokumentowany (<https://docs.pytest.org>)
- Spora liczba rozszerzeń

INSTALACJA

```
>>> pip install pytest
```

PIERWSZY TEST

```
def reciprocal(x):  
    return 1/x  
  
def test_first():  
    assert reciprocal(1) == 2
```

https://github.com/ifar/tech3camp-pytest/blob/master/tests/operations/test_1_first_test.py
https://github.com/ifar/tech3camp-pytest/blob/master/tests/operations/test_2_first_test.py

PARAMETRIZACJA TESTÓW

```
@pytest.mark.parametrize("x, y", (  
    [1, 1],  
    [0, "error"],  
    [-2, -0.5]  
))  
  
def test_reciprocal(x, y):  
    assert reciprocal(x) == y
```

https://github.com/ifar/tech3camp-pytest/blob/master/tests/operations/test_3_parametrize.py

https://github.com/ifar/tech3camp-pytest/blob/master/tests/operations/test_4_parametrize.py

OZNACZANIE TESTÓW (TAGOWANIE)

```
@pytest.mark.negative  
def test_reciprocal_0():  
    assert reciprocal(0) == "error"
```

```
pytest test_operations.py -m negative
```

https://github.com/ifar/tech3camp-pytest/blob/master/tests/operations/test_5_marks.py

SKIP / SKIPIF

```
@pytest.mark.skip(reason="Not yet implemented")
def test_reciprocal_0():
    assert reciprocal(0) == "error"

@pytest.mark.skipif(sys.version_info < (3, 8),
                    reason="requires python3.8 or higher")
def test_sqrt_minus_1():
    assert sqrt(-1) == "error"
```

https://github.com/ifar/tech3camp-pytest/blob/master/tests/operations/test_6_skip.py

XFAIL

```
@pytest.mark.xfail(reason="Issue #1225: Reciprocal of zero is not handled")
def test_reciprocal_0():
    assert reciprocal(0) == "error"

@pytest.mark.xfail(reason="Issue #1224: Sqrt of zero is not handled",
                  strict=True)
def test_sqrt_minus_1():
    assert sqrt(-1) == "error"
```

https://github.com/ifar/tech3camp-pytest/blob/master/tests/operations/test_7_xfail.py

FIXTURE

```
@pytest.fixture
def basic_request():
    url = "https://api.exchangeratesapi.io/2010-01-12"
    params = {'base': 'USD',
              'symbols': 'PLN',
              }
    proxy = {'http': '',
            'https': ''}
    w = Waluty()
    data = w.get_request(url, params, proxy)
    return data

def test_usd_to_pln(basic_request):
    usd_to_pln = basic_request['rates']['PLN']
    assert usd_to_pln == 2.8201091085
```

https://github.com/ifar/tech3camp-pytest/blob/master/tests/waluty/test_1_fixtures.py

https://github.com/ifar/tech3camp-pytest/blob/master/tests/waluty/test_2_fixture2.py

XDIST PLUGIN

```
pytest test.py -n 20
```

Instalacja: <https://pypi.org/project/pytest-xdist/>

```
pip install pytest-xdist
```

PLIK KONFIGURACYJNY PYTEST

- `conftest.py`:
- kod (np. `fixture`) jest uruchamiany przed testami
- `conftest.py` może znajdować się na różnych poziomach (w różnych podkatalogach)
- Opis: [link](#)

PYTEST - CIEKAWOSTKI

- parametryzacja
- oznaczanie parametrów specjalnymi znacznikami
- własny sposób wyświetlania nazw Przypadków Testowych (ids function)
- formatowanie / refactoring kodu przypadków testowych - w celu łatwiejszego ich utrzymania

https://github.com/ifar/tech3camp-pytest/blob/master/tests/operations/test_9_interesting.py

INTEGRACJA TESTÓW AUTOMATYCZNYCH Z TOX

```
pip install flake8  
pip install mypy  
pip install tox
```

<https://github.com/ifar/tech3camp-pytest/blob/master/tox.ini>



Dziękuję za uwagę 😊



Rafał Błaczkowski
Deep Learning Software Engineer
Internet of Things Group

<https://github.com/ifar/tech3camp-pytest>