# pytest-scenario Documentation

***Release 1.0a1***

**Ori Menashe**

May 15, 2016

Contents

pytest-scenario is a *pytest* plugin that aims to extend current test parameterization capabilities. After installing pytest-scenario you will be able run a test suite constructed from a JSON formatted test plan (AKA Test Scenario).

**Note**: pytest-scenario is currently classified as *alpha*, feel free to contact me with any issue at: https://github.com/OriMenashe

CHAPTER 1

# Features

- Test parameterization (including fixtures and test arguments).
- Test instantiation - run multiple test instances with different parameters.
- Test ordering - running tests in a thoughtful, user-defined order.
- Test exclusion - excluding unwanted tests during collection stage.

CHAPTER 2

# Installation

Install pytest-scenario by running:

```
pip install pytest-scenario
```

CHAPTER 3

# Quickstart

- Test parameterization is done by using a new **test_case** marker as follows:

```
@pytest.fixture
def db1(request):
    # connect to db1 database server...
    return db1

@pytest.fixture
def db2(request):
    # connect to db2 database server...
    return db2

class TestDBIntegrity:

    @pytest.mark.test_case(fixture_binding=[('db', 'db1', 'session')], params=[('table', 'USERS'), (
    def test_value_exists(self, db, table, field_name, field_value):
            # Query USERS table inside db1 for a user named orim.
            assert db.query("SELECT * from {} WHERE '{}'=={};").format(table, field_name, field_value)
```

- Test scenario is a JSON file located at the root of your project at <projects_root>/sut/scenarios/**<scenario_name>.json** :

```
[
    {
        "id": 1,
        "module_name": "tests.db_tests",
        "class_name": "TestDBIntegrity",
        "test_name": "test_value_exists",
        "fixture_binding": {
            "db": [
                "db1",
                "session"
            ]
        },
        "params": {
            "table": "USERS",
            "field_name": "user_name",
            "field_value": "orim"
        },
        "skip": false,
        "xfail": false
    },
    {
```

```
        "id": 2,
        "module_name": "tests.db_tests",
        "class_name": "TestDBIntegrity",
        "test_name": "test_value_exists",
        "fixture_binding": {
            "db": [
                "db2",
                "session"
            ]
        },
        "params": {
            "table": "USERS",
            "field_name": "user_name",
            "field_value": "miked"
        },
        "skip": false,
        "xfail": false
    },
]
```

Invocation of a test scenario will be done as follows:

```
~/workspace/projects_root$ py.test tests/ --scenario=<scenario_name>
```

CHAPTER 4

# License

The project is licensed under the MIT license.