



Become QA Auto

Фікстури

Бутенко Сергій





План лекції



Що таке фікстури



Практика

◎ **Fixtures**



Фікстури - це функції, які виконуються Pytest до фактичного виконання тестового сценарію та, якщо це потрібно, після нього.



Проблема



```
test_api.py X
tests > api > test_api.py > ...
You, seconds ago | 1 author (You)
1 class User:
2
3     def __init__(self) -> None:
4         self.name = 'Sergii'
5         self.second_name = 'Butenko'
6
7
8 user = User()
9
10
11 def test_remove_name():
12     user.name = ''
13     assert user.name == ''
14
15
16 def test_name():
17     assert user.name == 'Sergii'
18
19
20 def test_second_name():
21     assert user.second_name == 'Butenko'
22
23 You, seconds ago • Uncommitted changes
```



Проблема



```
→ Become QA Auto git:(master) X pytest
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto
plugins: Faker-4.1.1, rerunfailures-9.1.1, forked-1.3.0, anyio-3.4.0, xdist-1.32
.0, allure-pytest-2.8.16, html-2.1.1, metadata-1.10.0, ordering-0.6, instafail-0
.4.2
collected 3 items

tests/api/test_api.py .F. [100%]

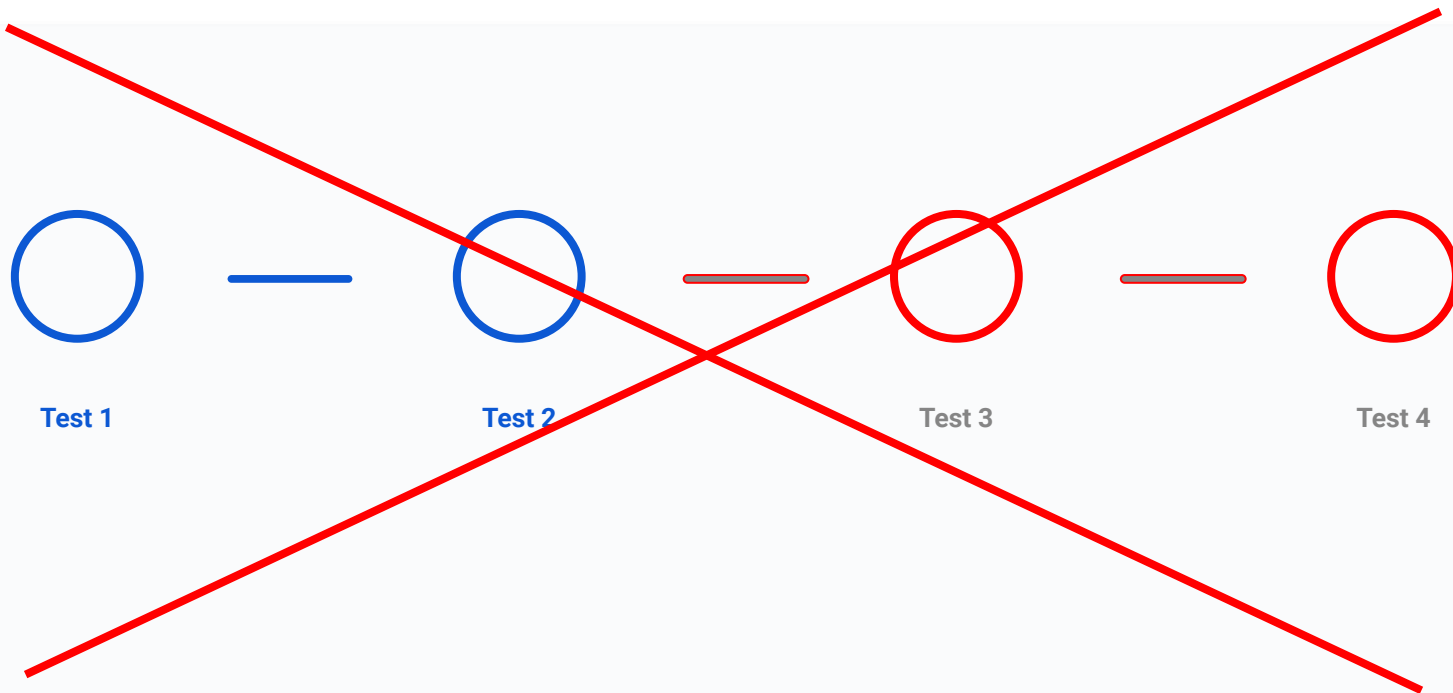
===== FAILURES =====
_____ test_name _____

    def test_name():
>       assert user.name == 'Sergii'
E       AssertionError: assert '' == 'Sergii'
E       - Sergii

tests/api/test_api.py:17: AssertionError
===== short test summary info =====
FAILED tests/api/test_api.py::test_name - AssertionError: assert '' == 'Sergii'
===== 1 failed, 2 passed in 0.07s =====
```



Рішення





Рішення



The screenshot shows the VS Code interface. On the left is the Explorer sidebar with the file tree expanded to 'tests > api > __pycache__'. The main editor window shows the file 'test_api.py' with the following code:

```
1 import pytest
2
3
4 class User:
5
6     def __init__(self) -> None:
7         self.name = 'Sergii'
8         self.second_name = 'Butenko'
9
10
11 @pytest.fixture
12 def user():
13     yield User()
14
15
```

At the bottom of the editor, a status bar indicates 'You, seconds ago • Uncommitted changes'.



Рішення



```
15
16 def test_remove_name(user):
17     user.name = ''
18     assert user.name == ''
19
20
21 def test_name(user):
22     assert user.name == 'Sergii'
23
24
25 def test_second_name(user):
26     assert user.second_name == 'Butenko'
27
```




Рішення



```
→ Become QA Auto git:(master) X pytest
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto
plugins: Faker-4.1.1, rerunfailures-9.1.1, forked-1.3.0, anyio-3.4.0, xdist-1.32
.0, allure-pytest-2.8.16, html-2.1.1, metadata-1.10.0, ordering-0.6, instafail-0
.4.2
collected 3 items

tests/api/test_api.py ... [100%]

===== 3 passed in 0.07s =====
→ Become QA Auto git:(master) X
```



Проблема



OPEN EDITORS

- BECOME QA AUTO
 - .pytest_cache
 - config
 - config.py
 - modules
 - api / clients
 - __init__.py
 - common
 - __init__.py
 - ui / page_objects
 - __init__.py
 - tests
 - api
 - __pycache__
 - test_api.py
 - test_fixture... 1, U
 - ui
 - __pycache__
 - test_ui.py

tests > api > test_fixtures.py > ...

```
1 class User:
2
3     def __init__(self) -> None:
4         self.name = None
5         self.second_name = None
6
7     def create(self):
8         self.name = 'Sergii'
9         self.second_name = 'Butenko'
10
11     def remove(self):
12         self.name = ''
13         self.second_name = ''
14
15
```



Проблема



```
api
├── __pycache__
├── test_api.py
└── test_fixtures.py
ui
├── __pycache__
└── test_ui.py
```

```
15
16 def test_change_name():
17     user = User()
18     user.create()
19
20     assert user.name == 'Sergii'
21     user.remove()
22
23
24 def test_change_second_name():
25     user = User()
26     user.create()
27
28     assert user.second_name == 'Butenko'
29     user.remove()
30
```



Проблема



```
→ Become QA Auto git:(master) X pytest
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto
plugins: Faker-4.1.1, rerunfailures-9.1.1, forked-1.3.0, anyio-3.4.0, xdist-1.32
.0, allure-pytest-2.8.16, html-2.1.1, metadata-1.10.0, ordering-0.6, instafail-0
.4.2
collected 5 items

tests/api/test_api.py ... [ 60%]
tests/api/test_fixtures.py .. [100%]

===== 5 passed in 0.04s =====
→ Become QA Auto git:(master) X
```



Проблема



```
api
├── __pycache__
├── test_api.py
└── test_fixtures.py
ui
├── __pycache__
└── test_ui.py
```

```
15
16 def test_change_name():
17     user = User()
18     user.create()
19
20     assert user.name == 'Sergii'
21     user.remove()
22
23
24 def test_change_second_name():
25     user = User()
26     user.create()
27
28     assert user.second_name == 'Butenko'
29     user.remove()
30
```



Проблема



```
OPEN EDITORS
BECOME QA AUTO
  .pytest_cache
  config
    config.py
  modules
    api / clients
      __init__.py
    common
      __init__.py
    ui / page_objects
      __init__.py
  tests
    api
      > __pycache__
      test_api.py
      test_fixture... 1, U
    ui
      > __pycache__
      test_ui.py

tests > api > test_fixtures.py > ...
1  import pytest
2
3
4  class User:
5
6      def __init__(self) -> None:
7          self.name = None
8          self.second_name = None
9
10     def create(self):
11         self.name = 'Sergii'
12         self.second_name = 'Butenko'
13
14     def remove(self):
15         self.name = ''
16         self.second_name = ''
17
18
19 @pytest.fixture
20 def user():
21     user = User()
22     user.create()
23
24     yield user
25
26     user.remove()
```



Проблема



```
28
29 def test_change_name(user):
30     assert user.name == 'Sergii'
31
32
33 def test_change_second_name(user):
34     assert user.second_name == 'Butenko'
35
```



Проблема



```
→ Become QA Auto git:(master) X pytest
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto
plugins: Faker-4.1.1, rerunfailures-9.1.1, forked-1.3.0, anyio-3.4.0, xdist-1.32
.0, allure-pytest-2.8.16, html-2.1.1, metadata-1.10.0, ordering-0.6, instafail-0
.4.2
collected 5 items

tests/api/test_api.py ... [ 60%]
tests/api/test_fixtures.py .. [100%]

===== 5 passed in 0.03s =====
→ Become QA Auto git:(master) X
```




Підсумки



- ✓ **Фікстури** - це функції, які виконуються Pytest до фактичного виконання тестового сценарію та, якщо це потрібно, після нього.
- ✓
 - **Перша проблема**, яку вирішують фікстури - це залежність тестів один від одного, якщо вони використовують спільні дані
- ✓
 - **Друга проблема**, яку вирішують фікстури - перевикористання коду і виконання коду до та після тесту