

# Як тестувати API. Пишемо свій API клієнт

Бутенко Сергій

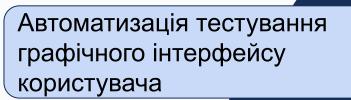
## Властивості автоматизації тестування





Швидкість





Автоматизація тестування API

Автоматизація тестування коду



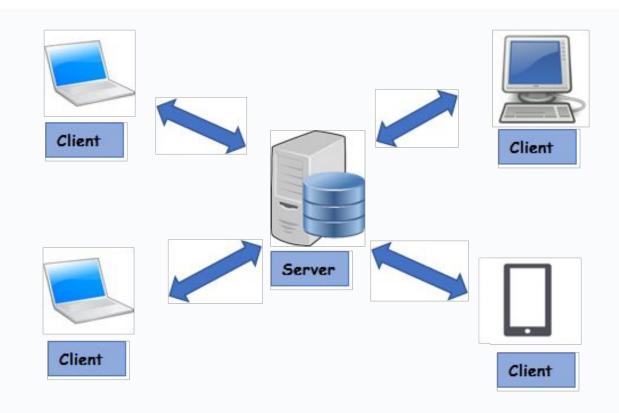
Час на розробку





# Клієнт-серверна архітектура





## АРІ тестування. Задача

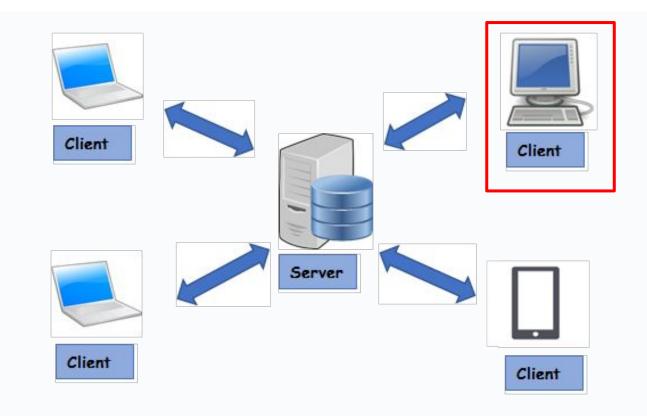


#### Постановка задачі

Зробити клієнт, який би використовував GITHUB API для комунікації з сервером та використати його в тесті

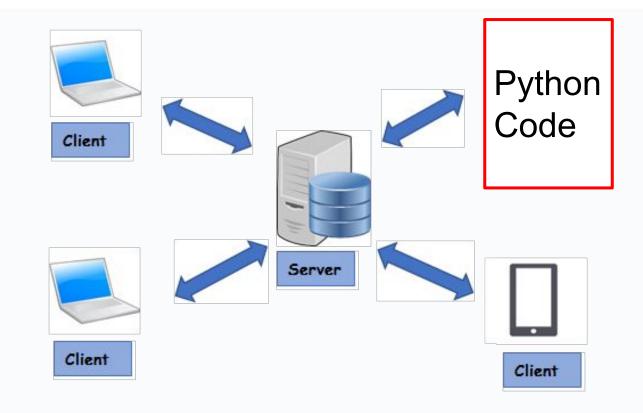
# **О АРІ тестування**





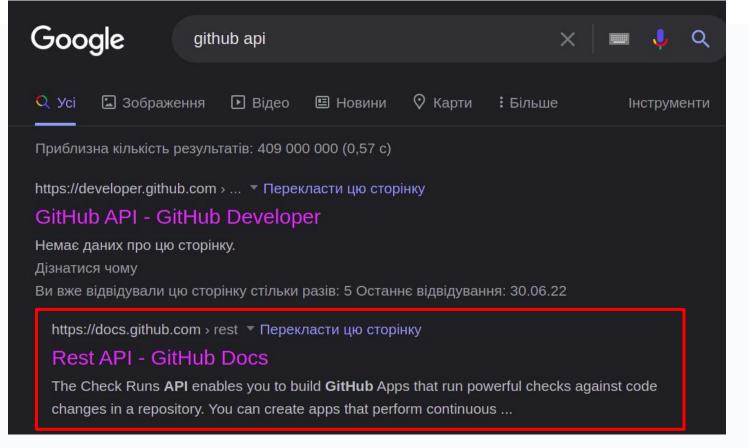
# **О АРІ тестування**



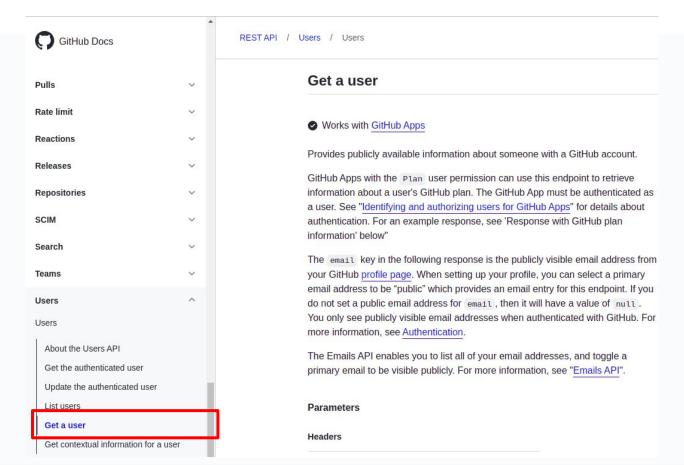


#### Практика











#### Get a user

Works with GitHub Apps

Provides publicly available information about someone with a GitHub account.

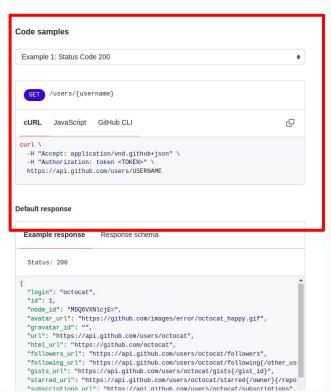
GitHub Apps with the Plan user permission can use this endpoint to retrieve information about a user's GitHub plan. The GitHub App must be authenticated as a user. See "Identifying and authorizing users for GitHub Apps" for details about authentication. For an example response, see 'Response with GitHub plan information' below"

The email key in the following response is the publicly visible email address from your GitHub profile page. When setting up your profile, you can select a primary email address to be "public" which provides an email entry for this endpoint. If you do not set a public email address for email, then it will have a value of null. You only see publicly visible email addresses when authenticated with GitHub. For more information, see Authentication.

The Emails API enables you to list all of your email addresses, and toggle a primary email to be visible publicly. For more information, see "Emails API".

#### **Parameters**

# Headers accept string Setting to application/vnd.github+json is recommended. Path parameters username string Required The handle for the GitHub user account.





# Parameters Headers accept string Setting to application/vnd.github+json is recommended. Path parameters username string Required The handle for the GitHub user account.

#### HTTP response status codes

Status code	Description	
200	ОК	
404	Resource not found	

#### Default response

```
Example response
                     Response schema
Status: 200
"login": "octocat",
"id": 1,
"node_id": "MDQ6VXNlcjE=",
"avatar url": "https://qithub.com/images/error/octocat happy.gif",
"gravatar_id": "",
"url": "https://api.github.com/users/octocat",
"html url": "https://github.com/octocat",
"followers_url": "https://api.github.com/users/octocat/followers",
"following_url": "https://api.github.com/users/octocat/following{/other_users/octocat/following}
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}",
"starred_url": "https://api.github.com/users/octocat/starred{/owner}{/repo
"subscriptions url": "https://api.github.com/users/octocat/subscriptions",
"organizations_url": "https://api.github.com/users/octocat/orgs",
"repos_url": "https://api.github.com/users/octocat/repos",
"events_url": "https://api.github.com/users/octocat/events{/privacy}",
"received events url": "https://api.github.com/users/octocat/received even
"type": "User",
"site admin": false,
"name": "monalisa octocat",
```



#### **Parameters**

# Headers accept string Setting to application/vnd.github+json is recommended. Path parameters username string Required The handle for the GitHub user account.

HTTP response status codes

Status code Description

404 Resource not found

OK

200

#### Default response

```
Example response
                     Response schema
 Status: 200
"login": "octocat",
"id": 1,
"node_id": "MDQ6VXNlcjE=",
"avatar_url": "https://github.com/images/error/octocat_happy.gif",
"gravatar_id": "",
"url": "https://api.github.com/users/octocat",
"html url": "https://github.com/octocat",
"followers_url": "https://api.github.com/users/octocat/followers",
"following_url": "https://api.github.com/users/octocat/following{/other_users/octocat/following}
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}",
"starred_url": "https://api.github.com/users/octocat/starred{/owner}{/repo
 "subscriptions url": "https://api.github.com/users/octocat/subscriptions",
 "organizations url": "https://api.github.com/users/octocat/orgs",
"repos_url": "https://api.github.com/users/octocat/repos",
"events_url": "https://api.github.com/users/octocat/events{/privacy}",
"received events url": "https://api.github.com/users/octocat/received even
"type": "User",
 "site admin": false,
"name": "monalisa octocat",
```

#### Практика



```
EXPLORER
                             test http.py X
                       ...
                             tests > api > dest_http.py > 😚 test_second_request
> OPEN EDITORS
                                    import pytest

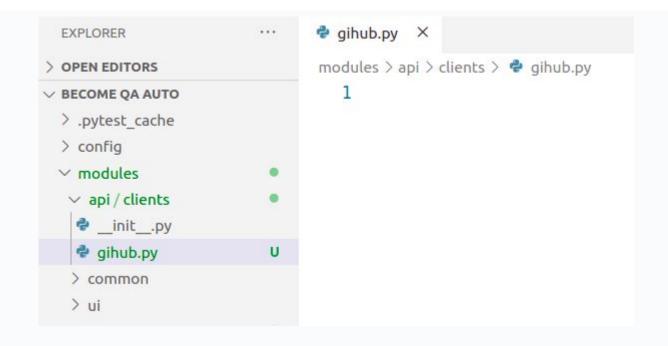
∨ BECOME QA AUTO

                                    import requests
 > .pytest cache
 > config
 > modules
                                    @pytest.mark.http
 v tests
                                    def test first request():
  > pycache
                                        r = requests.get('https://api.github.com/zen')
                                        print(f"Response is {r.text}")

√ api

   > pycache
                                    @pytest.mark.http
   test api.py
                                    def test second request():
   test fixtures.py
                                        r = requests.get('https://api.github.com/users/defunkt')
                               12
   test http.py
                      1, U
                               13
                                        body = r.json()
  ∨ ui
                                        headers = r.headers
                               14
                               15
   > pycache
                                        assert body['name'] == 'Chris Wanstrath'
                               16
   test ui.py
                              17
                                        assert r.status code == 200
  conftest.py
                                        assert headers['Server'] == 'GitHub.com'
                               18
 pytest.ini
                               19
                                                                                                          12
                               20
```







```
gihub.py X
                                               test_http.py
 EXPLORER
                               modules > api > clients > 💠 gihub.py > ...
> OPEN EDITORS
                                      import requests
∨ BECOME QA AUTO
 > .pytest_cache
  > config
                                      class GitHub:
 ∨ modules

→ api / clients

                                  6
                                           def get user defunkt(self):
   __init__.py
                                                r = requests.get('https://api.github.com/users/defunkt')
                                 8
                                               body = r.json()
   gihub.py
                         U
                                 9
   > common
                                                return body
                                10
   > ui
                                11
```







```
EXPLORER
 OPEN EDITORS
∨ BECOME QA AUTO
  > __pycache__

∨ .pytest cache

   > v
  .gitignore
   ≡ CACHEDIR.TAG
  (i) README.md
  > config
  > modules
 ∨ tests
   > __pycache__
   ∨ api
    > _pycache_
    test api.py
    test fixtures.py
                           U
    test github api.py
```

```
test github api.py X
tests > api > dest github api.py > test user exists
       import pytest
       from modules.api.clients.gihub import GitHub
       @pytest.mark.api
       def test user exists():
           api = GitHub()
           user = api.get user defunkt()
  8
           assert user['login'] == 'defunkt'
  9
 10
```



```
→ ~ cd repos/LnD/Become\ QA\ Auto
→ Become OA Auto git:(master) X pytest -m api
platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
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 10 items / 9 deselected / 1 selected
tests/api/test github api.py .
                                                          100%
```

#### Опрактика



```
→ Become QA Auto git:(master) X pytest -m api
   platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
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
collected 8 items / 1 error / 8 deselected / 0 selected
               ======== info ====== short test summary info
ERROR tests/api/test github_api.py
!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!
```



```
→ ~ cd repos/LnD/Become\ QA\ Auto
→ Become OA Auto git:(master) X pytest -m api
platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
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 10 items / 9 deselected / 1 selected
tests/api/test github api.py .
                                                          100%
```



```
gihub.py X
 EXPLORER
                             test github api.py
                             modules > api > clients > @ gihub.py > ...
OPEN EDITORS
                                    import requests
BECOME QA AUTO
 > pycache

∨ .pytest cache

                                    class GitHub:
  > v
 .gitignore
                                         def get user defunkt(self):
                                6
  ≡ CACHEDIR.TAG
                                             r = requests.get('https://api.github.com/users/defunkt')
                                8 9
                                             body = r.json()
 (i) README.md
 > config
                               10
                                             return body

∨ modules

                               11
  > pycache
                               12
                                         def get non exist user(self):

→ api / clients

                                             r = requests.get('https://api.github.com/users/butenkosergii')
                               13
  > __pycache__
                                             body = r.json()
                               14
  init .py
                               15
                                             return body
                               16
  gihub.py
                        U
                               17
```

## Практика



```
test github api.py X 🕏 test http.py
                                                                     gihub.py
 EXPLORER
                             tests > api > dest_github_api.py > test_user_not_exists
 OPEN EDITORS
                                     import pytest

∨ BECOME QA AUTO

                                     from modules.api.clients.gihub import GitHub
 > pycache
 > .pytest cache
 > config
                                    @pytest.mark.api
 > modules
                                     def test user exists():

∨ tests

                                         api = GitHub()
                                         user = api.get user defunkt()
  pycache
                                         assert user['login'] == 'defunkt'
                                9
  ∨ api
                               10
   pycache
                               11
   test api.py
                               12
                                    @pytest.mark.api
   test fixtures.py
                               13
                                     def test user not exists():
   test github api.py
                      2, U
                                         api = GitHub()
                               14
                                         r = api.get non exist user()
   test_http.py
                       1, U
                               15
```



```
EXPLORER
> OPEN EDITORS

∨ BECOME QA AUTO

  > pycache
  > .pytest cache
  > config
  > modules
 v tests
   > pycache
  ∨ api
    > pycache
   test_api.py
                          M
    test fixtures.py
                           U
   test github api.py
                         1, U
   test http.py
                           U
   > ui
```

```
test github api.py X e conftest.py
                                    gihub.py
tests > api > dest github api.py > test user not exists
      import pytest
      from modules.api.clients.gihub import GitHub
      @pytest.mark.api
      def test user exists():
           api = GitHub()
           user = api.get user defunkt()
           assert user['login'] == 'defunkt'
 10
 11
 12
      @pytest.mark.api
      def test user not exists():
 13
           api = GitHub()
 14
 15
           r = api.get non exist user()
           print(r)
 16
```



```
→ Become QA Auto git:(master) X pytest -s -m api
olatform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
olugins: Faker-4.1.1, rerunfailures-9.1.1, forked-1.3.0, anyio-3.4.0, xdist-1.32
.O, allure-pytest-2.8.16, html-2.1.1, metadata-1.10.0, ordering-0.6, instafail-0
4.2
collected 10 items / 8 deselected / 2 selected
tests/api/test github api.py {'message': 'Not Found', 'documentation url': 'htt
s://docs.github.com/rest/reference/users#get-a-user'}
```



```
→ Become OA Auto git:(master) X pytest -s -m api
olatform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
olugins: 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 10 items / 8 deselected / 2 selected
ests/api/test_github_api.py {'message': 'Not Found', 'documentation_url': 'htt:
s://docs.github.com/rest/reference/users#get-a-user'}
                :===== 2 passed. 8 deselected in 0.58s ====
```

#### Практика



```
EXPLORER
 OPEN EDITORS

∨ BECOME OA AUTO

  > pycache
  > .pytest cache
 > config
 > modules
 v tests
   > pycache
  ∨ api
    > pycache
   test_api.py
                          M
   test_fixtures.py
   test github api.py
                        1, U
   test http.py
                          U
   > ui
 conftest.py
                          U
 = pytest.ini
                          U
```

```
aihub.py
test_github_api.py X
                     conftest.py
tests > api > dest github api.py > test user not exists
      import pytest
      from modules.api.clients.gihub import GitHub
      @pytest.mark.api
      def test user exists():
           api = GitHub()
          user = api.get user defunkt()
           assert user['login'] == 'defunkt'
 10
 11
      @pytest.mark.api
 12
      def test user not exists():
 13
 14
           api = GitHub()
 15
           r = api.get non exist user()
           assert r['message'] == 'Not Found'
 16
```

#### Опрактика



```
Become OA Auto git:(master) / pytest -m api
platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become OA Auto, configfile: pytest.ini
plugins: Faker-4.1.1, rerunfailures-9.1.1, forked-1.3.0, anyio-3.4.0, xdist-1.32
.O, allure-pytest-2.8.16, html-2.1.1, metadata-1.10.0, ordering-0.6, instafail-0
.4.2
collected 10 items / 8 deselected / 2 selected
tests/api/test github api.py ...
```



```
conftest.py
                                                                     gihub.py X
  EXPLORER
                              test github api.py
                              modules > api > clients > 🔮 gihub.py > ...
> OPEN EDITORS
                                     import requests
∨ BECOME QA AUTO
  > pycache
  > .pytest cache
                                     class GitHub:
 > config

∨ modules

                                          def get user defunkt(self):
  > pycache
                                              r = requests.get('https://api.github.com/users/defunkt')
                                              body = r.json()
                                 8

→ api / clients

   > __pycache__
                                              return body
                               10
   __init__.py
                                11
   gihub.py
                         U
                                          def get non exist user(self):
                                12
   > common
                                              r = requests.get('https://api.github.com/users/butenkosergii')
                                13
  ) ui
                                              body = r.json()
                               14
  init .py
                         U
                               15
                                              return body
                               16
                         0

∨ tests
```



```
test github api.py conftest.py
 EXPLORER
                                                                     gihub.py X
                              modules > api > clients > ♥ gihub.py > ❤ GitHub > ♥ get user
 OPEN EDITORS
                                     import requests

∨ BECOME QA AUTO

 > pycache
 > .pytest_cache
                                     class GitHub:
 > config
                                 5

∨ modules

                                          def get_user(self, username):
  > __pycache__
                                              r = requests.get(f'https://api.github.com/users/{username}')
                                              body = r.json()

→ api / clients

                                 8
   > _pycache_
                                              return body
                                10
   init .py
   gihub.py
                       1, U
```



```
test github api.py conftest.py
 EXPLORER
                                                                     gihub.py X
                              modules > api > clients > ♥ gihub.py > ❤ GitHub > ♥ get user
 OPEN EDITORS
                                     import requests

∨ BECOME QA AUTO

 > pycache
 > .pytest_cache
                                     class GitHub:
 > config

∨ modules

                                         def get user(self username):
  > __pycache__
                                              r = requests.get(f'https://api.github.com/users/{username}'
                                              body = r.json()

→ api / clients

   > _pycache_
                                              return body
                                10
   init .py
   gihub.py
                       1, U
```



```
test github api.py X gihub.py
                                                                   test api.py
 EXPLORER
                             tests > api > 🕏 test github api.py > ...
> OPEN EDITORS
                                    import pytest
∨ BECOME QA AUTO
                                    from modules.api.clients.gihub import GitHub
 > pycache
 > .pytest cache
 > config
                                    @pytest.mark.api
 ∨ modules
                                    def test user exists():
  > pycache
                                         api = GitHuh()
                                         user = api.get user('defunkt')

→ api / clients

                                9
                                         assert user[ login ] == defunkt
   > __pycache__
                               10
   init .py
                               11
   aihub.py
                       1, U
                               12
                                    @pytest.mark.api
  > common
                                    def test user not exists():
                               13
  ) ui
                                         api - GitHub()
                               14
                                         r = api.get user('butenkosergii')
  init .py
                               15
                               16
                                         assert r['message'] == 'Not Found
 ∨ tests
                               17
  > __pycache__
  ∨ api
   pycache
   test api.py
                        M
   test fixtures.py
                        U
   test github api.py
                        U
```



```
test github api.py
                                                 conftest.py X
                                                                 gihub.py
 EXPLORER
                            conftest.py > ...
> OPEN EDITORS
                                  import pytest
∨ BECOME QA AUTO
                                   from modules.api.clients.gihub import GitHub
 > pycache
 > .pytest cache
 > config
                                   class User:
 ∨ modules
                              6
  > pycache
                                       def init (self) -> None:
                              8
                                           self.name = None

y api / clients

                                           self.second name = None
   > pycache
                             10
   init .py
                                       def create(self):
                             11
   gihub.py
                     1. U
                                           self.name = 'Sergii'
                             12
  > common
                                           self.second name = 'Butenko'
                             13
  > ui
                             14
                                       def remove(self):
  init_.py
                             15
                                           self.name = ''
                             16
 ∨ tests
                                           self.second name = ''
                             17
  > pycache
                             18
  ∨ api
                             19
   > pycache
                                  @pytest.fixture
                             20
   test api.py
                             21
                                  def user():
   test fixtures.py
                             22
                                       user = User()
   test github api.py
                                      user.create()
   test http.py
                                     PROBLEMS 1
                            TERMINAL
                                                  DEBUG CONSOLE
                                                                OUTPUT
  > ui
 conftest.py
 pytest.ini
```



```
conftest.py X
                                                                    gihub.py
 EXPLORER
                             test_github_api.py
> OPEN EDITORS
                              conftest.py > ...
                               14

∨ BECOME QA AUTO

                                         def remove(self):
                               15
 > pycache
                                              self.name = ''
                               16
 > .pytest cache
                                              self.second name = ''
                               17
 > config
                               18
                               19
 ∨ modules
                                    @pytest.fixture
                               20
  > pycache
                                    def user():

→ api / clients

                               22
                                         user = User()
   > _pycache_
                               23
                                         user.create()
   __init__.py
                               24
   aihub.py
                      1, U
                                         yield user
                               25
  > common
                               26
                               27
                                         user.remove()
  ) ui
                               28
  init .py
                               29
 ∨ tests
                                    @pytest.fixture
                               30
  > __pycache__
                               31
                                    def github api():
  ∨ api
                               32
                                         api = GitHub()
   > pycache
                               33
                                         yield api
   test api.py
                               34
                        M
   test fixtures.py
                        U
   test_github_api.py
                        U
   test http.py
                                       PROBLEMS 1
                              TERMINAL
                                                     DEBUG CONSOLE OUTPUT
  > ui
 conftest.py
                        U
 pvtest.ini
                        U
```



```
EXPLORER
OPEN EDITORS

✓ BECOME QA AUTO

 > pycache
 > .pytest_cache
 > config
 > modules

∨ tests

   > pycache
  ∨ api
    ) pycache
   test_api.py
   test fixtures.py
   test github api.py
   test http.py
                           U
```

```
test github api.py X e conftest.py
                                     gihub.py
tests > api > dest github api.py > ...
      import pytest
      @pytest.mark.api
      def test user exists(github api):
           user = github api.get user('defunkt')
           assert user['login'] == 'defunkt'
  9
 10
      @pytest.mark.api
      def test user not exists(github api):
 11
           r = github api.get user('butenkosergii')
 12
           assert r['message'] == 'Not Found'
 13
 14
```



```
EXPLORER
 OPEN EDITORS

✓ BECOME QA AUTO

 > pycache
 > .pytest_cache
 > config
 > modules

∨ tests

   > pycache
  ∨ api
    > pycache
   test_api.py
   test fixtures.py
   test github api.py
   test http.py
                           U
```

```
test github_api.py X
                    conftest.py
                                     gihub.py
tests > api > dest github api.py > ...
      import pytest
      @pytest.mark.api
      def test user exists(github api):
           user = github_api.get_user('defunkt')
           assert user['login'] == 'defunkt'
 10
      @pytest.mark.api
      def test user not exists(github api):
           r = github api.get user('butenkosergii')
 12
           assert r['message'] == 'Not Found'
 13
 14
```



```
EXPLORER
 OPEN EDITORS

✓ BECOME QA AUTO

 pycache
 > .pytest cache
 > config
 > modules

∨ tests

   > pycache
  ∨ api
    ) pycache
   test_api.py
   test fixtures.py
   test github api.py
   test http.py
                           U
```

```
conftest.py
                                     gihub.py
test github api.py X
tests > api > dest github api.py > ...
       import pytest
      @pytest.mark.api
      def test user exists(github api):
           user = github api.get user('defunkt')
           assert user['login'] == 'defunkt'
  8
  9
 10
      @pytest.mark.api
       def test user not exists(github api):
           r = github api.get user('butenkosergii')
 12
           assert r['message'] == 'Not Found'
 13
 14
```



```
→ Become OA Auto git:(master) X pytest -m api
platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become OA Auto, configfile: pytest.ini
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 10 items / 8 deselected / 2 selected
tests/api/test github api.py ...
                                                                100%
          ======= 2 passed. 8 deselected in 0.68s =====
```

#### Практика



```
import pytest
     from modules.api.clients.gihub import GitHub
                                                           import pytest
     @pytest.mark.api
                                                           @pytest.mark.api
     def test user exists():
                                                           def test user exists(github api):
         api = GitHub()
                                                               user = github api.get user('defunkt')
                                                               assert user['login'] == 'defunkt'
         user = api.get user defunkt()
         assert user['login'] == 'defunkt'
10
                                                           @pytest.mark.api
11
                                                           def test user not exists(github api):
     @pytest.mark.api
12
                                                               r = github api.get user('butenkosergii')
     def test user not exists():
13
                                                               assert r['message'] == 'Not Found'
         api = GitHub()
14
         r = api.get non exist user()
15
         assert r['message'] == 'Not Found'
16
```



```
@pytest.mark.http
10
     def test second request():
         r = requests.get('https://api.github.com/users/defunkt')
12
         body = r.json()
13
         headers = r.headers
14
15
         assert body['name'] == 'Chris Wanstrath'
16
         assert r.status code == 200
17
         assert headers['Server'] == 'GitHub.com'
18
19
20
     @pytest.mark.http
21
     def test status code request():
22
         r = requests.get('https://api.github.com/users/sergii butenko')
23
24
         assert r.status code == 404
25
26
```



```
import pytest
    @pytest.mark.api
    def test user exists(github api):
         user = github api.get user('defunkt')
         assert user['login'] == 'defunkt'
    @pytest.mark.api
10
    def test user not exists(github api):
11
         r = github_api.get_user('butenkosergii')
12
         assert r['message'] == 'Not Found'
13
14
```

## **Підсумки**



- Навчилися:
  - 1. **Створювати** власний АРІ клієнт на базі АРІ документації
  - 2. Використовувати АРІ клієнт в тестах
- З'ясували що використанні АРІ клієнту в тестах допомагає нам в:
  - 1. Зменшенні кількості повторювального кода в тестах
  - 2. Спрощує подальшу підтримку тестів
  - 3. Дозволяє **ефективно використовувати** фікстури в тестах