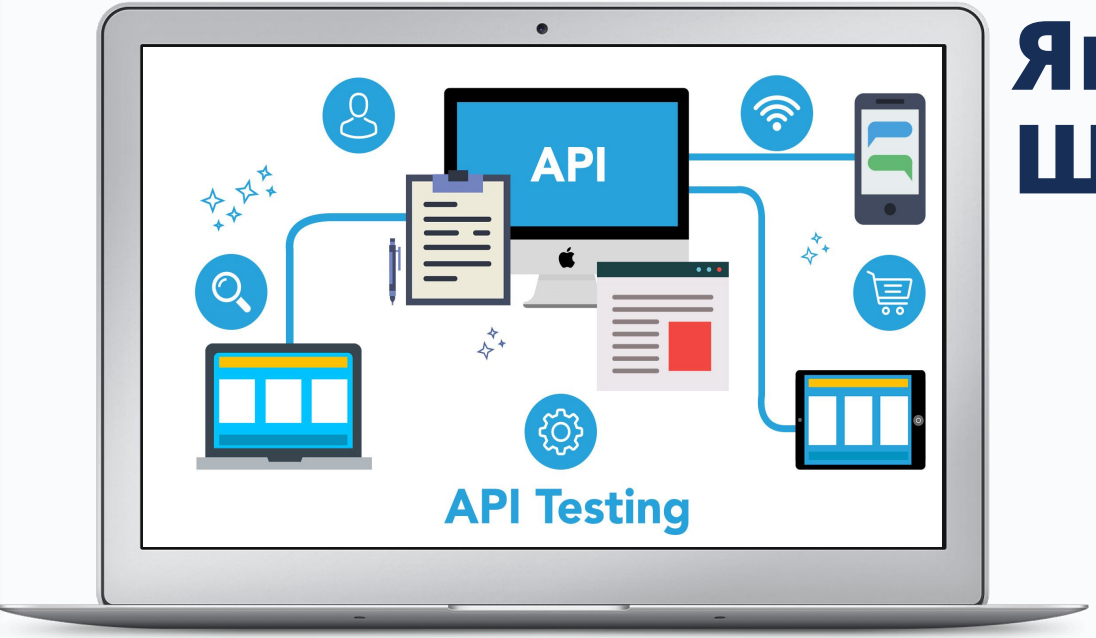
PROMETHEUS



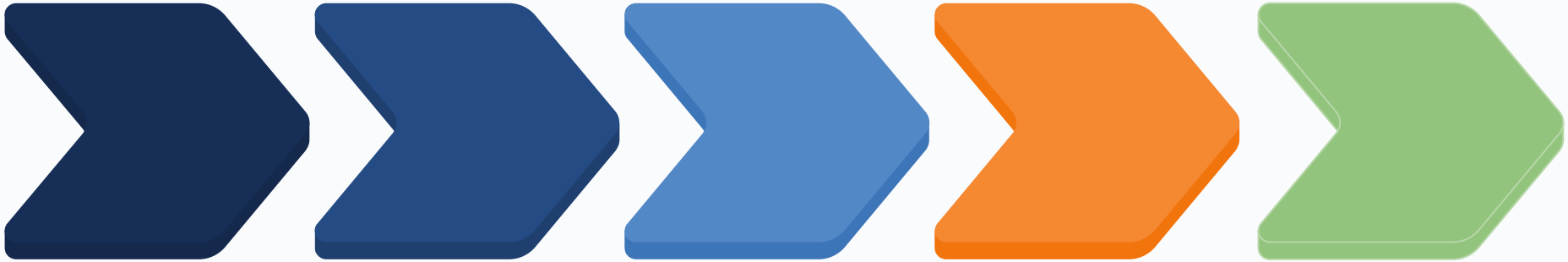
Як тестувати API. Ще тести

Бутенко Сергій

План лекції

- Опис робочого процесу
- Практика

◎Робочий процес



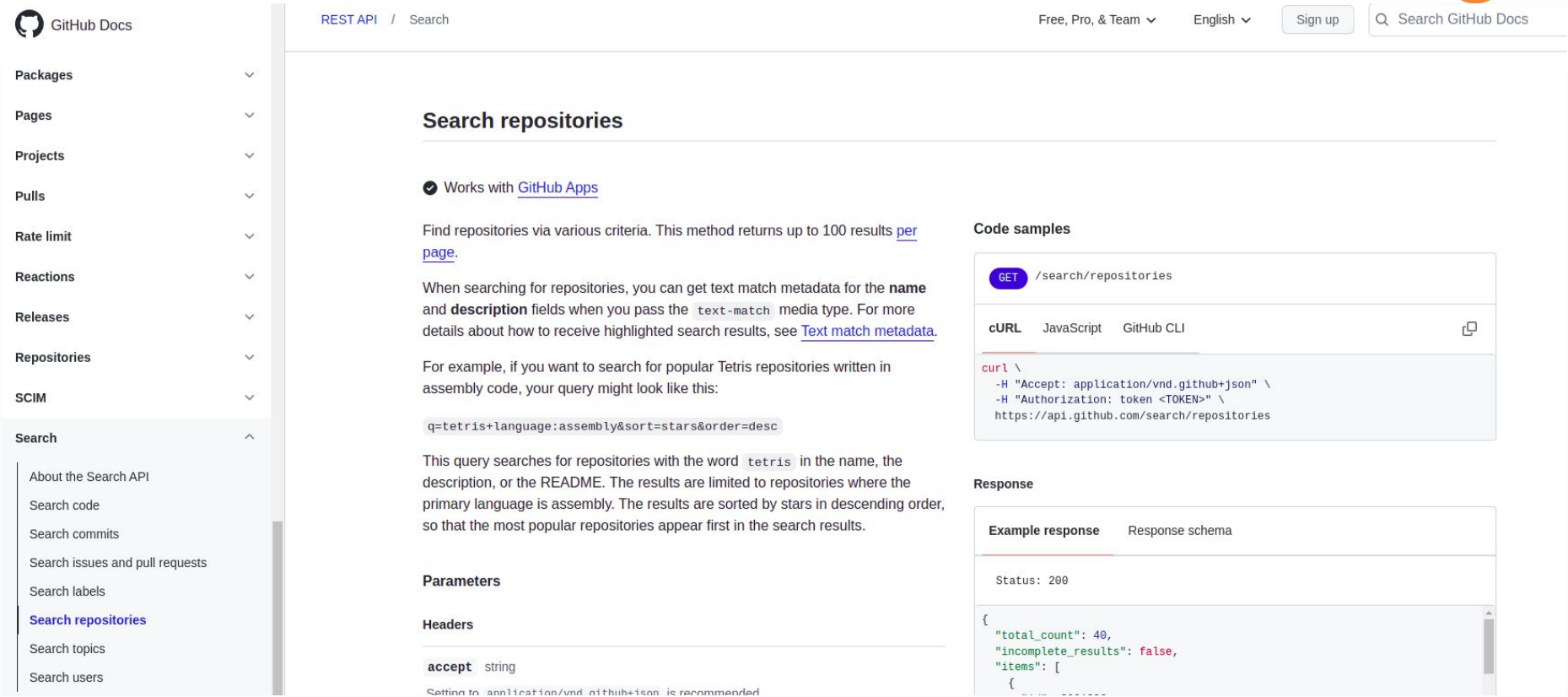
- Постановка задачі
- Аналіз залежностей
- Написання тестів
- Запуск тестів
- Звіт

◎Постановка задачі

◎Аналіз залежностей

A	B	C	D
ID	**TestCase Summary**	**Steps To Reproduce**	**Expected Result**
1	User is able to search for an existing repo	1. Send HTTP request with existing repository name in parameters	Repository found
2	User is able to search for non existing repo	1. Send HTTP request with non existing repository name in parameters	Repository not found
3	User is able to seach for the repo with name that consists from 1 character	1. Send HTTP request with existing repository name with length of 1 char	No error occured

Аналіз залежностей



Аналіз залежностей

Search repositories

Works with GitHub Apps

Find repositories via various criteria. This method returns up to 100 results per page.

When searching for repositories, you can get text match metadata for the **name** and **description** fields when you pass the `text-match` media type. For more details about how to receive highlighted search results, see Text match metadata.

For example, if you want to search for popular Tetris repositories written in assembly code, your query might look like this:

```
q=tetris+language:assembly&sort=stars&order=desc
```

This query searches for repositories with the word `tetris` in the name, the description, or the README. The results are limited to repositories where the primary language is assembly. The results are sorted by stars in descending order, so that the most popular repositories appear first in the search results.

Code samples

GET /search/repositories

cURL | JavaScript | GitHub CLI

```
curl \
  -H "Accept: application/vnd.github+json" \
  -H "Authorization: token <TOKEN>" \
  https://api.github.com/search/repositories
```

Response

Example response | Response schema

Аналіз залежностей

Search repositories

Works with GitHub Apps

Find repositories via various criteria. This method returns up to 100 results per page.

When searching for repositories, you can get text match metadata for the **name** and **description** fields when you pass the `text-match` media type. For more details about how to receive highlighted search results, see Text match metadata.

For example, if you want to search for popular Tetris repositories written in assembly code, your query might look like this:

```
q=tetris+language:assembly&sort=stars&order=desc
```

This query searches for repositories with the word `tetris` in the name, the description, or the README. The results are limited to repositories where the primary language is assembly. The results are sorted by stars in descending order, so that the most popular repositories appear first in the search results.

Code samples

GET /search/repositories

cURL | JavaScript | GitHub CLI

```
curl \
  -H "Accept: application/vnd.github+json" \
  -H "Authorization: token <TOKEN>" \
  https://api.github.com/search/repositories
```

Response

Example response | Response schema

Аналіз залежностей

Search repositories

Works with GitHub Apps

Find repositories via various criteria. This method returns up to 100 results per page.

When searching for repositories, you can get text match metadata for the **name** and **description** fields when you pass the `text-match` media type. For more details about how to receive highlighted search results, see Text match metadata.

For example, if you want to search for popular Tetris repositories written in assembly code, your query might look like this:

```
q=tetris+language:assembly&sort=stars&order=desc
```

This query searches for repositories with the word `tetris` in the name, the description, or the README. The results are limited to repositories where the primary language is assembly. The results are sorted by stars in descending order, so that the most popular repositories appear first in the search results.

Code samples

GET /search/repositories

cURL | JavaScript | GitHub CLI

```
curl \
  -H "Accept: application/vnd.github+json" \
  -H "Authorization: token <TOKEN>" \
  https://api.github.com/search/repositories
```

Response

Example response | Response schema

Аналіз залежностей

Parameters

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Query parameters

`q` string — Required

The query contains one or more search keywords and qualifiers. Qualifiers allow you to limit your search to specific areas of GitHub. The REST API supports the same qualifiers as the web interface for GitHub. To learn more about the format of the query, see Constructing a search query. See "Searching for repositories" for a detailed list of qualifiers.

`sort` string

Sorts the results of your query by number of `stars`, `forks`, or `help-wanted-issues` or how recently the items were `updated`. Default: best match

Can be one of: `stars`, `forks`, `help-wanted-issues`, `updated`

Аналіз залежностей

Response

Example response | Response schema

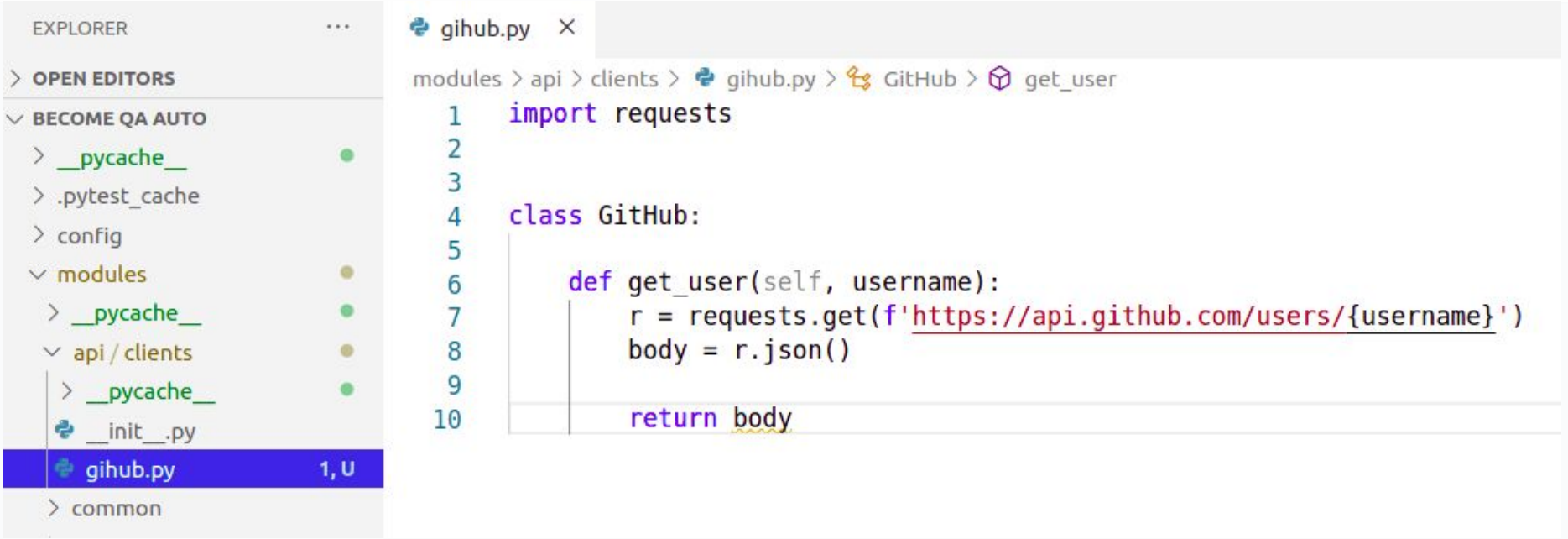
Status: 200


```
{
  "total_count": 40,
  "incomplete_results": false,
  "items": [
    {
      "id": 3081286,
      "node_id": "MDEwOlJlcG9zaXRvcnkzMDgxMjg2",
      "name": "Tetris",
      "full_name": "dtrupenn/Tetris",
      "owner": {
        "login": "dtrupenn",
        "id": 872147,
        "node_id": "MDQ6VXNlcjg3MjE0Nw==",
        "avatar_url": "https://secure.gravatar.com/avatar/e7956084e75f239de8
        "gravatar_id": "",
        "url": "https://api.github.com/users/dtrupenn",
        "received_events_url": "https://api.github.com/users/dtrupenn/receiv
        "type": "User",
        "html_url": "https://github.com/octocat",
        "followers_url": "https://api.github.com/users/octocat/followers",
```

Аналіз залежностей

A	B	C	D
ID	**TestCase Summary**	**Steps To Reproduce**	**Expected Result**
1	User is able to search for an existing repo	1. Send HTTP request with existing repository name in parameters	Repository found
2	User is able to search for non existing repo	1. Send HTTP request with non existing repository name in parameters	Repository not found
3	User is able to seach for the repo with name that consists from 1 character	1. Send HTTP request with existing repository name with length of 1 char	No error occured

Написання тестів



Написання тестів

EXPLORER

OPEN EDITORS

BECOME QA AUTO

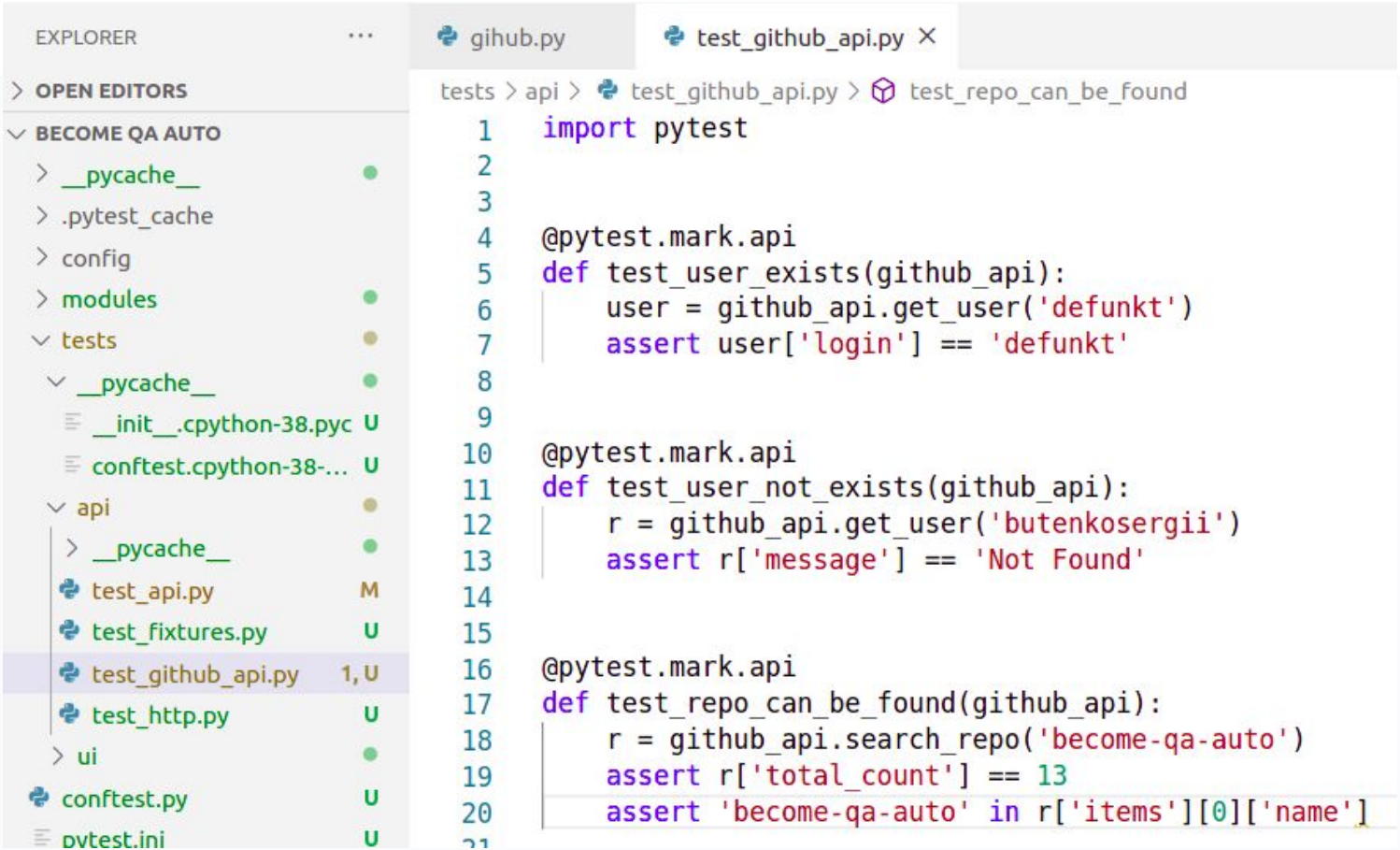
- __pycache__
- .pytest_cache
- config
- modules
  - __pycache__
  - api / clients
    - __pycache__
    - __init__.py
    - gihub.py U
  - common
  - ui
- __init__.py U
- tests
- conftest.py U
- pytest.ini U

gihub.py

modules > api > clients > gihub.py > ...

```python
import requests


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

        return body

    def search_repo(self, name):
        r = requests.get(
            "https://api.github.com/search/repositories",
            params={"q": name}
        )
        body = r.json()

        return body
```

Написання тестів

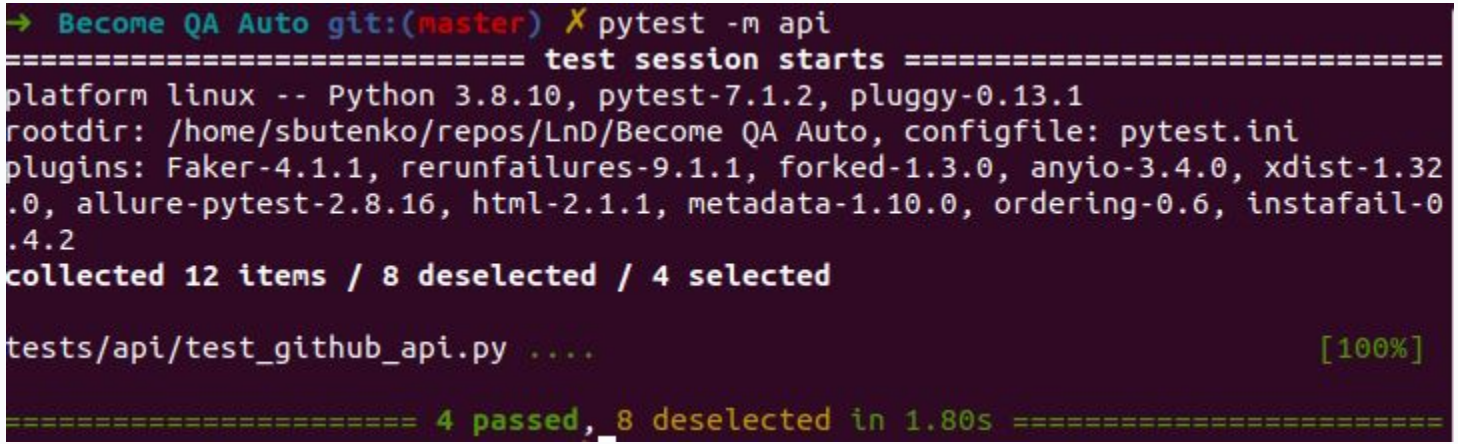


Написання тестів

```
→ Become QA Auto git:(master) ✗ pytest -m api
============================= 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, 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 11 items / 8 deselected / 3 selected

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

====================== 3 passed, 8 deselected in 1.19s =======================
```

Написання тестів

Написання тестів



Написання тестів

Написання тестів

```
→ Become QA Auto git:(master) ✗ pytest -m api
============================= 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, 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 13 items / 8 deselected / 5 selected

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

======================= 5 passed, 8 deselected in 3.13s ========================
```

Підсумки

- Написали тести в такому середовищі, з яким ви будете працювати на реальних проєктах
- Навчилися відправляти HTTP запити з параметрами пошукового рядка