**Лабораторна робота №6. Тестування API**

Мета: ознайомитись з методикою роботи з бібліотекою Supertest тестування API

**Теоретичні відомості**

Cучасне програмне забезпечення потребує використання програмних інтерфейсів HTTP API. В архітектурі мікросервісів вони з’єднують зовнішні та внутрішні або різні внутрішні компоненти. Широко використовуються API сторонніх розробників для основних функцій продукту, таких як оплата та зв’язок. Кожен API — це певний контракт, якого повинні дотримуватися обидві сторони, щоб залишатися на зв’язку. У випадку API це означає, що коди статусу, заголовки та тіла відповіді завжди повинні відповідати документації API.

**SuperTest (**[**https://github.com/ladjs/supertest**](https://github.com/ladjs/supertest)**)** бібліотеку тестування Node.js для HTTP API. Вона розширює іншу бібліотеку під назвою superagent, HTTP-клієнт JavaScript для Node.js і браузера. Розробники можуть використовувати SuperTest як окрему бібліотеку або з фреймворками тестування JavaScript, такими як Mocha або Jest.

Інсталяція в проєкті: npm install supertest --save-dev.

Підключення в коді: const request = require('supertest');

Роблячи запит, можна ініціалізувати його двома різними способами. Можна протестувати зовнішній API або внутрішній API, який уже розгорнуто та запущено (принаймні в проміжному середовищі), надавши URL як параметр. Ось приклад створення зовнішнього запиту API. Використовуємо загальнодоступний Dog API (<https://dog.ceo/dog-api/>), який пропонує зображення собак із відкритим кодом. Це простий API, який не потребує реєстрації і, як такий, ідеально підходить для демонстрації використання зовнішніх API (рис. 1).

Метод get() повідомляє SuperTest, що використовується метод HTTP GET. Можна об’єднати додаткові методи у виклик для налаштування автентифікації, користувальницьких HTTP-заголовків, але це не обов’язково для цього прикладу. Метод end() завершує запит і викликає сервер API. Як параметр, для обробки відповіді потрібна функція зворотного виклику.

Альтернативний спосіб полягає у передачі об’єкта, якщо API із використовує Express. В останньому випадку SuperTest при необхідності запустить тестовий сервер і автоматично відправить туди запити. Ось приклад створення програми Express з однією кінцевою точкою (за мотивами Dog API), а потім надсилання запиту на цю саму кінцеву точку (рис. 2)

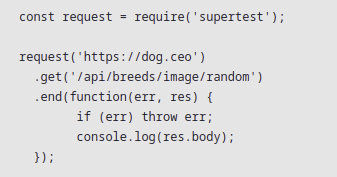


Рисунок 1 - Доступ до Dog API



Рисунок 2 - Приклад використання Express

Завдання для самостійної роботи

1. Ознайомитись з переліком API для тестування <https://gorest.co.in/>
2. Виконати тестування основних методів вибраного API.
3. Створити github репозиторій з тестами.

Контрольні запитання

1. Наведіть види тестування.
2. Особливості модульного та приймального тестування.
3. Призначення бібліотек Mocha та Chai.