Міністерство освіти і науки україни

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Математичний факультет

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Дисципліна «Автоматизоване тестування»

Лабораторна робота № 6

Виконав студент гр. 8.1214

Гавриш Іван Васильович

Перевірив: Викладач

Лебідь Максим Володимирович

Запоріжжя

2024

**Посилання на репозиторій:** [givtszs/automated-testing-lab6](https://github.com/givtszs/automated-testing-lab6)

**Завдання:**

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

**Хід роботи:**

1. Сайт https://gorest.co.in/ містить наступний перелік API доступних для тестування:



У ході виконання лабораторної роботи буде використано API для тестування кінцевої точки users.

1. Створимо набір тестів для API https://gorest.co.in/public/v2/users:

**const request = require('supertest');**

**const { assert } = require('chai');**

**describe('user API test', function () {**

**this.timeout(10000);**

**const apiUrl = 'https://gorest.co.in/public/v2';**

**const TOKEN = 'ab0ffeb1238fc44f2ede2d4e9ad44fc5d5c62f0d3999ef749eb76241aa864fb6';**

**const user = {**

**name: "John Doe",**

**email: "john\_doe@tester.com",**

**gender: "male",**

**status: "active"**

**};**

**let createdUserId;**

**it('GET should return a list of users', async () => {**

**await request(apiUrl)**

**.get('/users')**

**.expect(200);**

**});**

**it('POST should create a new user', async () => {**

**await request(apiUrl)**

**.post('/users')**

**.set('Authorization', `Bearer ${TOKEN}`)**

**.send(user)**

**.expect(201)**

**.expect((response) => {**

**createdUserId = response.body.id;**

**console.log(createdUserId);**

**assert.equal(response.body.name, user.name);**

**assert.equal(response.body.email, user.email);**

**assert.equal(response.body.gender, user.gender);**

**assert.equal(response.body.status, user.status);**

**});**

**});**

**it('GET should return user details', async () => {**

**await request(apiUrl)**

**.get(`/users/${createdUserId}`)**

**.set('Authorization', `Bearer ${TOKEN}`)**

**.expect(200)**

**.expect((response) => {**

**assert.equal(response.body.id, createdUserId);**

**assert.equal(response.body.name, user.name);**

**assert.equal(response.body.email, user.email);**

**assert.equal(response.body.gender, user.gender);**

**assert.equal(response.body.status, user.status);**

**});**

**});**

**it('PATCH should update user details', async () => {**

**const newName = 'Not John Doe';**

**await request(apiUrl)**

**.patch(`/users/${createdUserId}`)**

**.set('Authorization', `Bearer ${TOKEN}`)**

**.send({ name: newName })**

**.expect(200)**

**.expect((response) => {**

**assert.equal(response.body.name, newName);**

**});**

**});**

**it('DELETE should delete a user', async () => {**

**await request(apiUrl)**

**.delete(`/users/${createdUserId}`)**

**.set('Authorization', `Bearer ${TOKEN}`)**

**.expect(204);**

**});**

**});**

Виконаємо тести:

