

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашнее задание №4

Выполнил:

Борисова Элина

Группа К3341

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

- реализовать тестирование API средствами Postman;
- написать тесты внутри Postman.

Ход работы

В уже созданной коллекции напишем тесты для нескольких эндпоинтов:

```
pm.test("Status code is 200", function() {

    pm.response.to.have.status(200);

});

pm.test("Response is JSON", function() {

    pm.response.to.be.json;

});

pm.test("Response has correct structure", function() {

    const response = pm.response.json();

    pm.expect(response).to.be.an('object');

    pm.expect(response).to.have.all.keys('user_id', 'recipe_id');

    const uuidRegex =
/^([0-9a-f]{8}-[0-9a-f]{4}-4[0-9a-f]{3}-[89ab][0-9a-f]{3}-[0-9a-f]{12})$/i;

    pm.test("user_id is valid UUID", function() {

        pm.expect(response.user_id).to.be.a('string');

        pm.expect(response.user_id).to.match(uuidRegex);

    });
```

```
pm.test("recipe_id is valid UUID", function() {

    pm.expect(response.recipe_id).to.be.a('string');

    pm.expect(response.recipe_id).to.match(uuidRegex);

});

});

pm.test("Response matches request data", function() {

    const response = pm.response.json();

    const requestData = JSON.parse(pm.request.body.raw);

    pm.expect(response.user_id).to.eql(requestData.user_id);

    pm.expect(response.recipe_id).to.eql(requestData.recipe_id);

});

pm.test("Save data for future tests", function() {

    const response = pm.response.json();

    pm.environment.set("last_user_id", response.user_id);

    pm.environment.set("last_recipe_id", response.recipe_id);

});
```

HTTP

Recipe API / recipe-categories / Получить все связи рецептов с категориями

Save

GET

{{baseUrl}} /recipe-categories

S

Params

Authorization

Headers (6)

Body

Scripts

Settings

Pre-request

Post-response

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

pm.test("Status code is 200", function() {

pm.response.to.have.status(200);

});

pm.test("Response is JSON", function() {

pm.response.to.be.json;

});

pm.test("Response has correct structure", function() {

const response = pm.response.json();

pm.expect(response).to.be.an('array');

if (response.length > 0) {

pm.expect(response[0]).to.have.all.keys('id', 'name', 'description');

pm.expect(response[0].id).to.be.a('number');

pm.expect(response[0].name).to.be.a('string');

}

}

Packages

</> Snippets

Body

Cookies

Headers (7)

Test Results (4/4)

200 OK

34 ms

235 B

Save Re

Filter Results

PASSED

Status code is 200

PASSED

Response is JSON

PASSED

Response has correct structure

PASSED

Response time is acceptable

The screenshot shows the Postman interface for a POST request to the endpoint `{{baseUrl}} /likes`. The 'Scripts' tab is active, displaying a series of PM tests. The 'Test Results' tab at the bottom shows that all four tests passed successfully.

Tests:

- pm.test("Response is JSON", function() {
 pm.response.to.be.json;
});
- pm.test("Response has correct structure", function() {
 const response = pm.response.json();

 pm.expect(response).to.be.an('object');
 pm.expect(response).to.have.all.keys('user_id', 'recipe_id');

 const uuidRegex = /^[0-9a-f]{8}-[0-9a-f]{4}-4[0-9a-f]{3}-[89ab][0-9a-f]{3}-[0-9a-f]{12}\$/i;

 pm.test("user_id is valid UUID", function() {
 pm.expect(response.user_id).to.be.a('string')
 pm.expect(response.user_id).to.match(uuidRegex);
 });
});

Test Results (7/7):

- PASSED** Status code is 200
- PASSED** Response is JSON
- PASSED** user_id is valid UUID
- PASSED** recipe_id is valid UUID

Вывод

В ходе данной работы, были написаны тесты для запросов. Которые впоследствии были успешно пройдены.