

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет по домашней работе
Тестирование

Выполнил:

Елистратов В. Д.

Группа К33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Содержание отчета

1) Постановка задачи	3
2) Общий тест	3
2) Частные тесты	3
Вывод	5

1) Постановка задачи

Тестирование API из ЛР2 средствами Postman (написать 2-3 теста для конкретных эндпоинтов + 1 общий тест для всех эндпоинтов)


2) Общий тест

Проверяет время ответа сервера (не более 100 мс)

```
// default testing
pm.test('Server response time less than 50 ms', function () {
  pm.expect(pm.response.responseTime).to.be.below(50)
})
```

2) Частные тесты

Тест авторизации пользователя, получения токена и проверка валидности полученного токена



```
POST {{base_url}}/users/login

Params Authorization Headers (8) Body • Pre-request Script • Tests • Settings

1 function getJwtData (token) {
2   return JSON.parse(Buffer.from(token.split('.')[1], 'base64').toString())
3 }
4
5 // default testing
6 pm.test('Server response time less than 50 ms', function () {
7   pm.expect(pm.response.responseTime).to.be.below(50)
8 })
9
10 if (pm.response.code !== 200) {
11   pm.test('Login failure', function () {
12     pm.expect(pm.response.code).to.eql(200);
13   });
14 }
15 return
16 }
17
18 if (pm.response.code === 200) {
19   pm.test('Successful response', function () {
20     pm.expect(pm.response.code).to.eql(200);
21   });
22
23   pm.test('Token is valid', function () {
24     const requestEmail = JSON.parse(pm.request.body.raw)["email"]
25     const token = pm.response.json()["token"]
26     const tokenData = getJwtData(token)
27     pm.expect(tokenData["email"]).to.equal(requestEmail)
28
29     pm.environment.set("token", token)
30   })
31 }
32
33 postman.setNextRequest("Create comment")
34 }
35
```

POST Get token by authorization

http://localhost:3000/v1/users/login

- PASS Server response time less than 50 ms
- PASS Successful response
- PASS Token is valid

Создание нового комментария к пользователю

POST {{base_url}}/comments/user/add

Params Authorization Headers (9) Body Pre-request Script Tests Settings

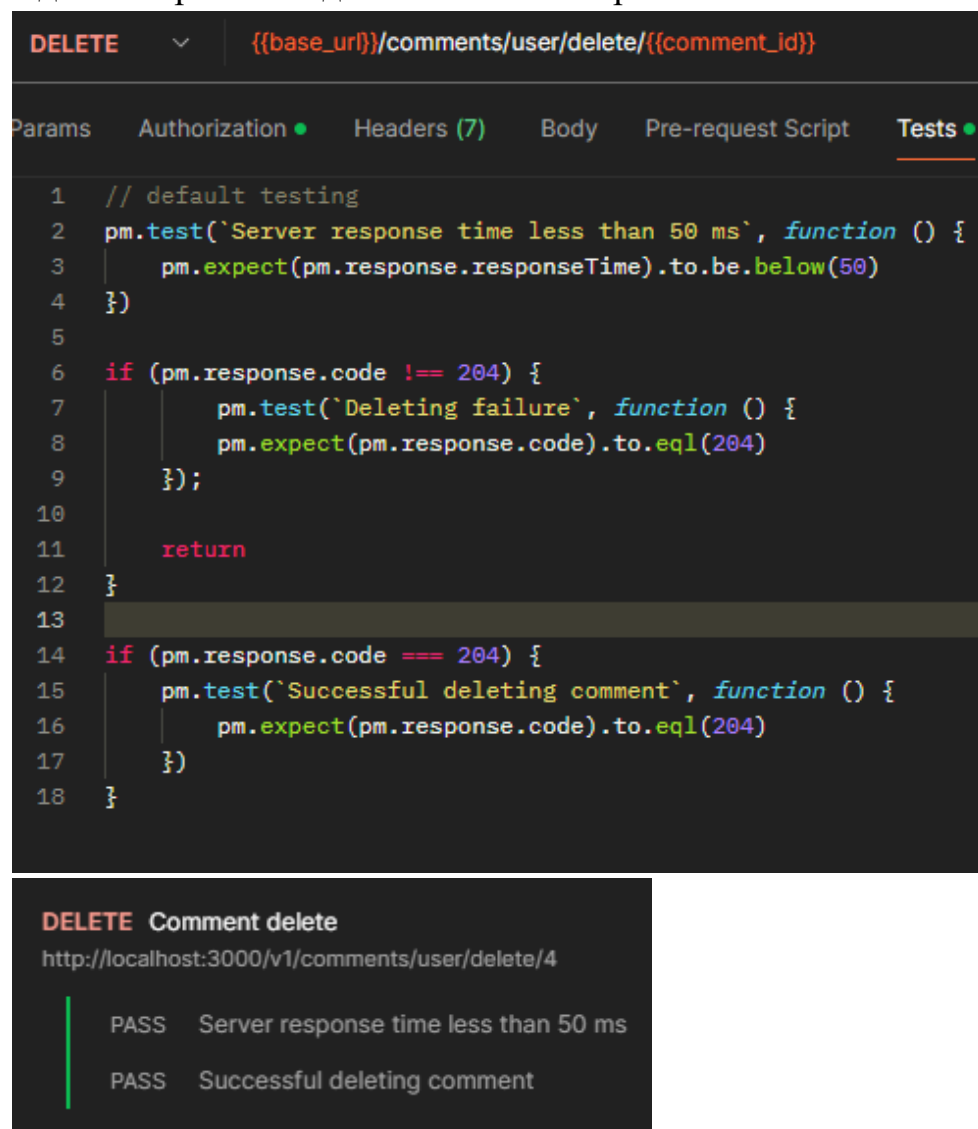
```
1 // default testing
2 pm.test('Server response time less than 50 ms', function () {
3   pm.expect(pm.response.responseTime).to.be.below(50)
4 })
5
6 if (pm.response.code !== 201) {
7   pm.test('Comment creation failed', function () {
8     pm.expect(pm.response.code).to.eql(201);
9   })
10
11   return
12 }
13
14 if (pm.response.code === 201) {
15   pm.test('Successful comment creation', function () {
16     pm.expect(pm.response.code).to.eql(201)
17   });
18
19   pm.test('Data is correct', function () {
20     const req = JSON.parse(pm.request.body.raw)
21     const res = pm.response.json()
22     pm.expect(req.title).to.eq(res["title"])
23     pm.expect(req.main_info).to.eq(res["main_info"])
24     pm.expect(req.rating).to.eq(0)
25
26     pm.environment.set("comment_id", res["id"])
27   })
28 }
29
30 postman.setNextRequest("Comment delete")
31 }
```

POST Create comment

http://localhost:3000/v1/comments/user/add

- PASS Server response time less than 50 ms
- PASS Successful comment creation
- PASS Data is correct

Удаление ранее созданного комментария



The screenshot displays the Postman interface for a DELETE request. The top bar shows the method 'DELETE' and the URL template '{{base_url}}/comments/user/delete/{{comment_id}}'. Below this, the 'Tests' tab is active, showing a JavaScript test script. The script includes a default test for response time, a failure test for non-204 status codes, and a success test for 204 status codes. Below the script, the 'Tests' section shows two passed tests: 'Server response time less than 50 ms' and 'Successful deleting comment'.

```
1 // default testing
2 pm.test('Server response time less than 50 ms', function () {
3   pm.expect(pm.response.responseTime).to.be.below(50)
4 })
5
6 if (pm.response.code !== 204) {
7   pm.test('Deleting failure', function () {
8     pm.expect(pm.response.code).to.eql(204)
9   });
10
11   return
12 }
13
14 if (pm.response.code === 204) {
15   pm.test('Successful deleting comment', function () {
16     pm.expect(pm.response.code).to.eql(204)
17   })
18 }
```

DELETE Comment delete
http://localhost:3000/v1/comments/user/delete/4

- PASS Server response time less than 50 ms
- PASS Successful deleting comment

Вывод

В домашней работе были написаны тесты для API из ЛР2 средствами Postman.