

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа 5

Выполнил:

Рыбалко Олег

Группа К33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

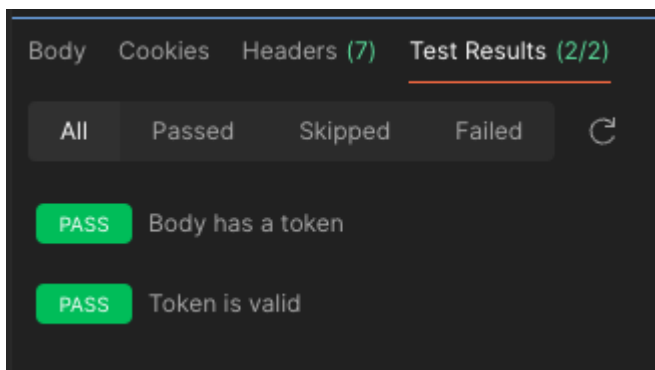
Тестирование API из ЛР2 средствами Postman (написать 2-3 теста для конкретных endpoint'ов + 1 общий тест для всех endpoint'ов)

Ход работы

1. Напишем тесты для авторизации. Для начала проверим, что от сервера приходит токен, а затем убедимся в его валидности

```
1 function parseJwt (token) {  
2   return JSON.parse(Buffer.from(token.split('.')[1], 'base64').toString());  
3 }  
4  
5 pm.test("Body has a token", function () {  
6   pm.expect(pm.response.json()).to.have.property("token")  
7 });  
8  
9 pm.test("Token is valid", function () {  
10   const payload = parseJwt(pm.variables.get("jwt_token"))  
11   const body = JSON.parse(pm.request.body.raw)  
12   pm.expect(payload.sub).to.equal(body.email)  
13   pm.expect(payload.exp).to.be.above(Math.floor(Date.now() / 1000))  
14 })
```

Выполним данный запрос и убедимся в том, что тесты проходят успешно

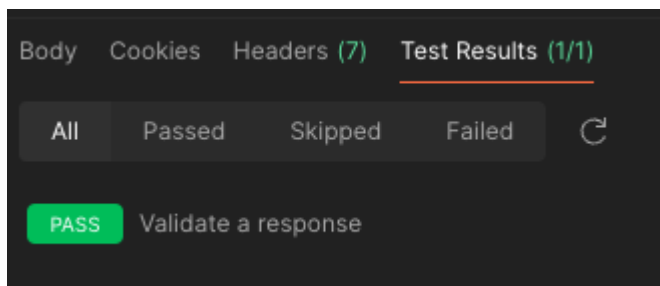


2. Напишем тест на удаление продукта. После выполнения запроса попытка получить удаленный продукт должна вернуть 404 ошибку

```
1 pm.test("Successfull delete", () => {  
2   const productID = pm.request.url.path[1]  
3   pm.sendRequest({  
4     method: "GET",  
5     header: {"Authorization": `Bearer ${pm.variables.get("jwt_token")}`},  
6     url: `${pm.variables.get("server_url")}/products/${productID}`, (err, res) => {  
7       pm.expect(res.code).to.eq(404)  
8     })  
9   pm.expect(pm.response.code).to.eq(204)  
10 })
```

3. Напишем тест на создания продукта, который проверит, что заданные поля правильно установлены

```
1 pm.test("Validate a response", () => {
2   const req = JSON.parse(pm.request.body.raw)
3   const res = JSON.parse(pm.response.stream.toString())
4   pm.expect(req.name).to.eq(res.name)
5   pm.expect(req.price).to.eq(res.price)
6   pm.expect(req.quantity).to.eq(res.quantity)
7   pm.expect(req.imageUrl).to.eq(res.imageUrl)
8 }
```



4. Напишем общий тест, который будет запускаться после выполнения каждого запроса в коллекции. Будем проверять, что токен еще не истек

```
1 function parseJwt (token) {
2   return JSON.parse(Buffer.from(token.split('.')[1], 'base64').toString());
3 }
4
5 pm.test("Token is valid", function () {
6   const payload = parseJwt(pm.variables.get("jwt_token"))
7   pm.expect(payload.exp).to.be.above(Math.floor(Date.now() / 1000))
8 })
```

Вывод

В данной домашней работе удалось написать тесты при помощи Postman для некоторых endpoint'ов из лабораторной работы №2.