

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа 5

Выполнил:

Пономарев Константин

Группа К33402

Проверил:

Добряков Д. И.

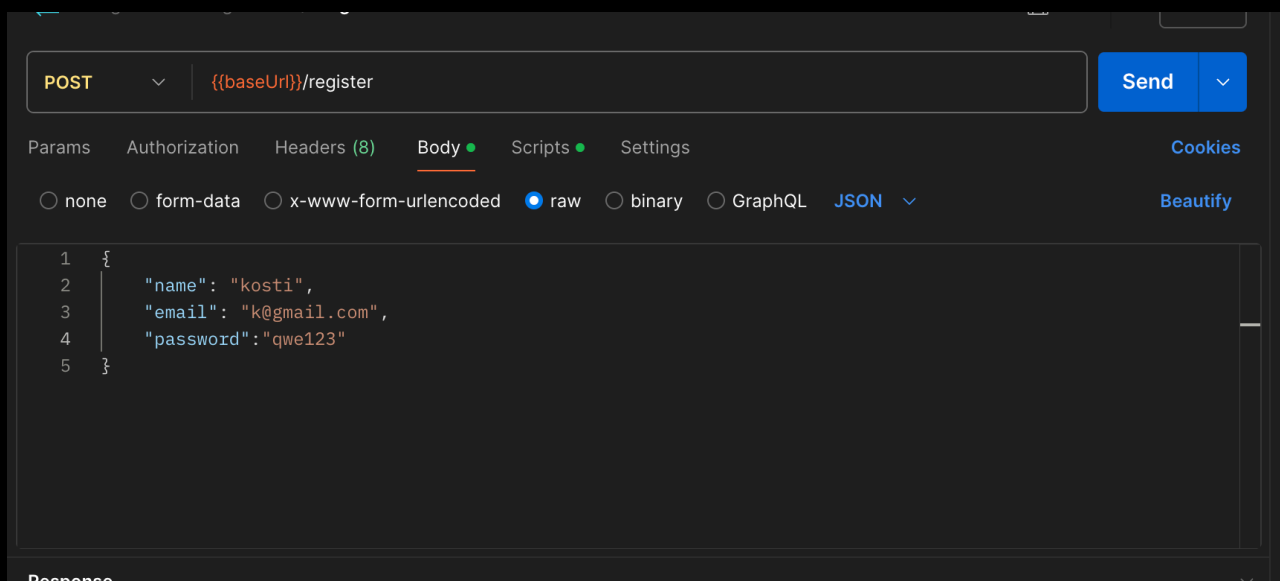
**Санкт-Петербург
2024 г.**

Задача:

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

Ход работы:

1) Регистрация юзера



Проверка статуса кода на 201 или 500 при ошибке валидации

Проверка наличия идентификатора пользователя, если статус-код 201

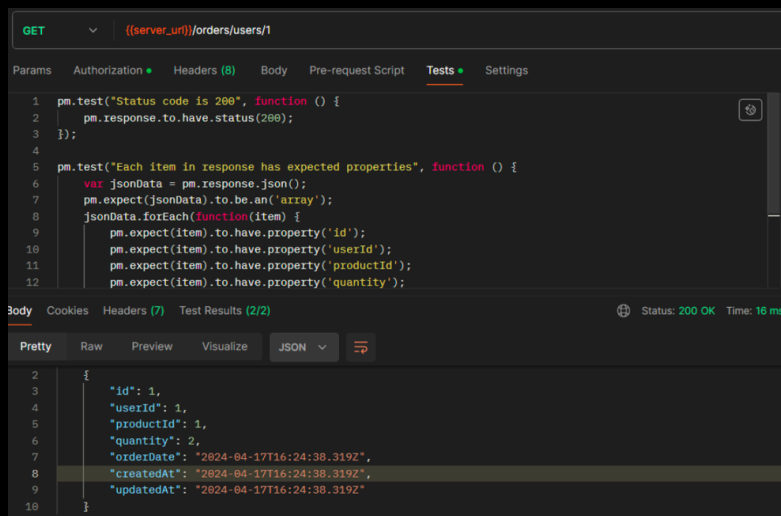
Проверка ошибки валидации, если пользователь уже существует

```

1
2 pm.test("Status code is 201 or 500 on validation error", function () {
3   pm.expect(pm.response.code).to.be.oneOf([201, 500]);
4   if (pm.response.code === 500) {
5     var jsonData = pm.response.json();
6     pm.expect(jsonData.message).to.eql("Validation Error");
7   }
8 });
9
10 pm.test("Response has user ID if status code is 201", function () {
11   if (pm.response.code === 201) {
12     var jsonData = pm.response.json();
13     pm.expect(jsonData).to.have.property('id');
14   }
15 });
16
17 pm.test("Validation Error if user already exists", function () {
18   if (pm.response.code === 500) {
19     var jsonData = pm.response.json();
20     pm.expect(jsonData.error).to.include("Validation Error");
21   }
22 });
23

```

2) Тест ручки получения заказа по userid



GET `{{server_url}}/orders/users/1`

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

```

1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("Each item in response has expected properties", function () {
6   var jsonData = pm.response.json();
7   pm.expect(jsonData).to.be.an('array');
8   jsonData.forEach(function(item) {
9     pm.expect(item).to.have.property('id');
10    pm.expect(item).to.have.property('userId');
11    pm.expect(item).to.have.property('productId');
12    pm.expect(item).to.have.property('quantity');
13

```

Body Cookies Headers (7) Test Results (2/2) Status: 200 OK Time: 16 ms

Pretty Raw Preview Visualize JSON

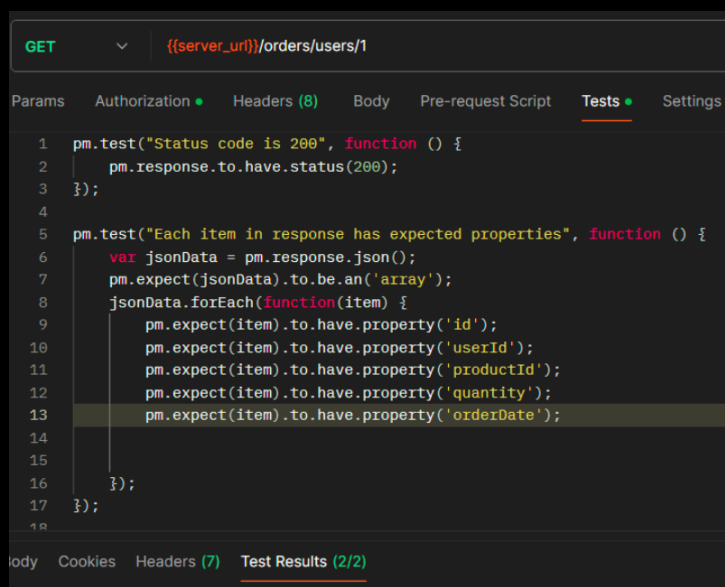
```

2 {
3   "id": 1,
4   "userId": 1,
5   "productId": 1,
6   "quantity": 2,
7   "orderDate": "2024-04-17T16:24:38.319Z",
8   "createdAt": "2024-04-17T16:24:38.319Z",
9   "updatedAt": "2024-04-17T16:24:38.319Z"
10 }

```

Проверка статус кода 200

Проверка наличия ожидаемых свойств в каждом элементе ответа



GET `{{server_url}}/orders/users/1`

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

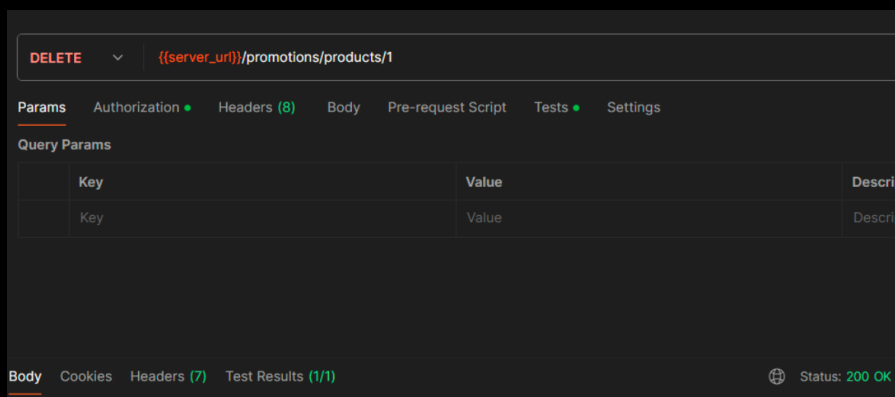
```

1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("Each item in response has expected properties", function () {
6   var jsonData = pm.response.json();
7   pm.expect(jsonData).to.be.an('array');
8   jsonData.forEach(function(item) {
9     pm.expect(item).to.have.property('id');
10    pm.expect(item).to.have.property('userId');
11    pm.expect(item).to.have.property('productId');
12    pm.expect(item).to.have.property('quantity');
13    pm.expect(item).to.have.property('orderDate');
14
15   });
16 });
17

```

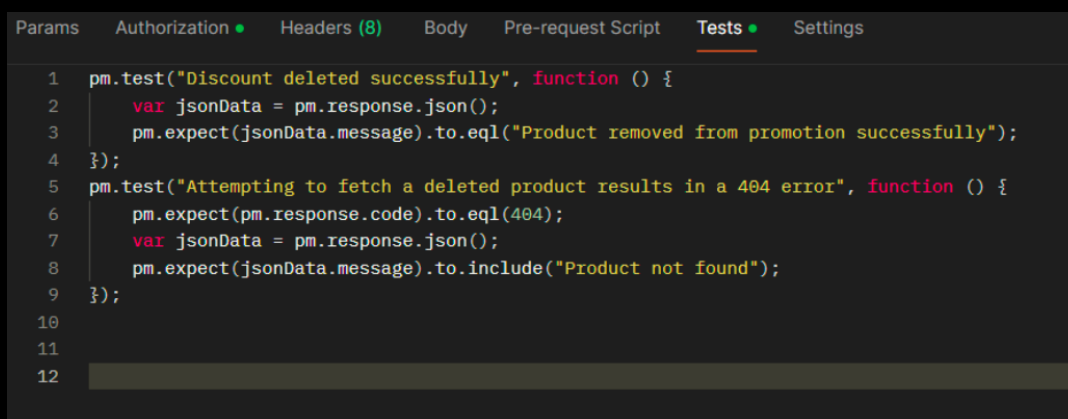
Body Cookies Headers (7) Test Results (2/2)

3) Тест ручки удаления товаров из акций

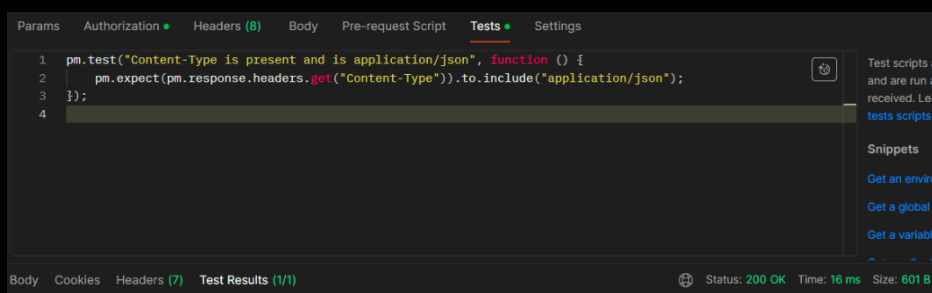


Тест на успешное удаление скидки

Тест на получение 404 при попытке доступа к удаленному продукту



проверяем content-type на application/json



Вывод: написал тестики на постмане, попробовал вообще что это такое. понял, что удобнее лично мне писать тесты непосредственно в проекте

