

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа №5
Тестирование API из ЛР2 средствами Postman

Выполнила:
Зайцева А. А.
Группа К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

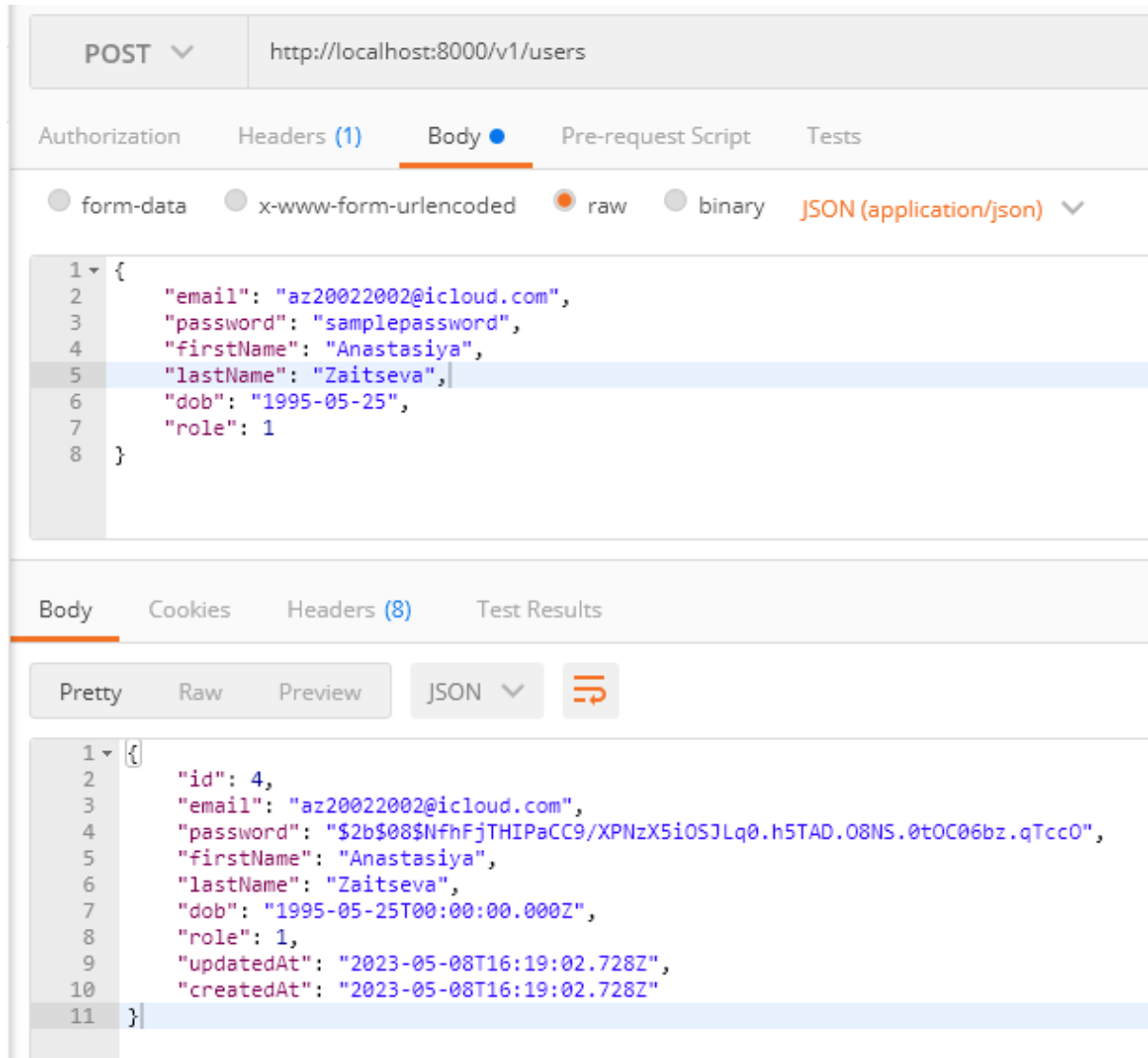
2023 г.

Задача

Тестирование API из ЛР2 средствами Postman

Ход работы

Создание нового пользователя



Вход в систему

POST http://localhost:8000/v1/users/login

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json) ▼

```
1 {
2   "email": "az20022002@icloud.com",
3   "password": "samplepassword"
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ▼

```
1 {
2   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NCwiaWF0IjoxNjgzNTYzMTg1fQ.-kQoQxhq13vS739BgkADyxw48ro1a_1iS2jMv0XPx-o",
3   "refreshToken": "61473668-26fc-4805-b296-51e182dc5ef0"
4 }
```

Получение данных текущего пользователя

GET http://localhost:8000/v1/users/profile

Authorization Headers (1) Body Pre-request Script Tests

| Key | Value |
|---|--|
| <input checked="" type="checkbox"/> Authorization | Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NCwiaWF0IjoxNjgzNTYzMTg1fQ.-kQoQxhq13vS739BgkADyxw48ro1a_1iS2jMv0XPx-o |
| New key | Value |

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ▼

```
1 {
2   "id": 4,
3   "email": "az20022002@icloud.com",
4   "password": "$2b$08$NfhFjTHIPaCC9/XPnzX5i0SJLq0.h5TAD.08NS.0t0C06bz.qTcc0",
5   "firstName": "Anastasiya",
6   "lastName": "Zaitseva",
7   "role": 1,
8   "dob": "1995-05-25T00:00:00.000Z",
9   "createdAt": "2023-05-08T16:19:02.728Z",
10  "updatedAt": "2023-05-08T16:19:02.728Z"
11 }
```

Получение данных пользователя по id

GET http://localhost:8000/v1/users/profile/3

Authorization Headers (1) Body Pre-request Script Tests

| Key | Value |
|---|---|
| <input checked="" type="checkbox"/> Authorization | Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NCwiaWF0IjoxNjgzNT... |
| New key | Value |

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON

```
1 {
2   "id": 3,
3   "email": "third@example.com",
4   "password": "Testtest123",
5   "firstName": "Third",
6   "lastName": "User",
7   "role": 3,
8   "dob": "2002-02-02T00:00:00.000Z",
9   "createdAt": "2023-05-08T12:58:54.373Z",
10  "updatedAt": "2023-05-08T12:58:54.373Z"
11 }
```

Обновление access токена по refresh токenu

POST http://localhost:8000/v1/users/refresh

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "refreshToken": "61473668-26fc-4805-b296-51e182dc5ef0"
3 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON

```
1 {
2   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NCwiaWF0IjoxNjgzNTYzNDA0fQ.4JIP6WYH_NM%ymjqHXKBiJ1nRmsHgkGrOhsAIUq3NT8",
3   "refreshToken": "8e2f2e8b-727e-4969-8e86-5ddeec4e50c4"
4 }
```

Получение всех товаров

GET

http://localhost:8000/v1/products

AuthorizationHeadersBodyPre-request ScriptTests

TypeNo Auth

BodyCookiesHeaders (8)Test Results

PrettyRawPreviewJSON

```
1  [
2    {
3      "id": 1,
4      "name": "Платье",
5      "price": 7999,
6      "createdAt": "2023-05-08T12:58:54.399Z",
7      "updatedAt": "2023-05-08T12:58:54.399Z"
8    },
9    {
10     "id": 2,
11     "name": "Шорты",
12     "price": 2399,
13     "createdAt": "2023-05-08T12:58:54.399Z",
14     "updatedAt": "2023-05-08T12:58:54.399Z"
15   },
16   {
17     "id": 3,
18     "name": "Блузка",
19     "price": 2599,
20     "createdAt": "2023-05-08T12:58:54.399Z",
21     "updatedAt": "2023-05-08T12:58:54.399Z"
22   },
23   {
24     "id": 4,
25     "name": "Пальто",
26     "price": 9999,
27     "createdAt": "2023-05-08T12:58:54.399Z",
28     "updatedAt": "2023-05-08T12:58:54.399Z"
29   },
30   {
31     "id": 5,
32     "name": "Пиджак",
33     "price": 6799,
34     "createdAt": "2023-05-08T12:58:54.399Z",
35     "updatedAt": "2023-05-08T12:58:54.399Z"
36   },
37   {
38     "id": 6,
39     "name": "Штаны",
40     "price": 3399
41   }
42 ]
```

Получение товара по id

GET

http://localhost:8000/v1/products/2

AuthorizationHeadersBodyPre-request ScriptTests

TypeNo Auth

BodyCookiesHeaders (8)Test Results

PrettyRawPreviewJSON

```
1  {
2    "id": 2,
3    "name": "Шорты",
4    "price": 2399,
5    "createdAt": "2023-05-08T12:58:54.399Z",
6    "updatedAt": "2023-05-08T12:58:54.399Z"
7  }
```

Добавление нового товара

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/v1/products`. The request body is a JSON object: `{ "name": "Панама", "price": 2499 }`. The response body is a JSON object: `{ "id": 7, "name": "Панама", "price": 2499, "updatedAt": "2023-05-08T16:33:08.524Z", "createdAt": "2023-05-08T16:33:08.524Z" }`.

```
POST http://localhost:8000/v1/products
```

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "name": "Панама",
3   "price": 2499
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ▼

```
1 {
2   "id": 7,
3   "name": "Панама",
4   "price": 2499,
5   "updatedAt": "2023-05-08T16:33:08.524Z",
6   "createdAt": "2023-05-08T16:33:08.524Z"
7 }
```

Обновление информации о товаре

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/v1/products`. The request body is a JSON object: `{ "id": 7, "price": 3000 }`. The response body is a JSON object: `{ "id": 7, "name": "Панама", "price": 3000, "updatedAt": "2023-05-08T16:34:17.688Z", "createdAt": "2023-05-08T16:33:08.524Z" }`.

```
POST http://localhost:8000/v1/products
```

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "id": 7,
3   "price": 3000
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ▼

```
1 {
2   "id": 7,
3   "name": "Панама",
4   "price": 3000,
5   "updatedAt": "2023-05-08T16:34:17.688Z",
6   "createdAt": "2023-05-08T16:33:08.524Z"
7 }
```

Удаление товара

DELETE ▼ http://localhost:8000/v1/products/7

Authorization Headers (1) Body ● Pre-request Script Tests

Type No Auth ▼

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ▼

1

Получение списка поставок

GET ▼ http://localhost:8000/v1/supplies/supply

Authorization Headers (1) Body Pre-request Script Tests

Type No Auth ▼

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ▼

```
1 [
2   {
3     "id": 1,
4     "quantity": 20,
5     "productId": 1,
6     "dateOfSupply": "2022-05-01T00:00:00.000Z",
7     "createdAt": "2023-05-08T12:58:54.407Z",
8     "updatedAt": "2023-05-08T12:58:54.407Z"
9   },
10  {
11    "id": 2,
12    "quantity": 5,
13    "productId": 1,
14    "dateOfSupply": "2022-05-02T00:00:00.000Z",
15    "createdAt": "2023-05-08T12:58:54.407Z",
16    "updatedAt": "2023-05-08T12:58:54.407Z"
17  },
18  {
19    "id": 3,
20    "quantity": 15,
21    "productId": 1,
22    "dateOfSupply": "2022-05-03T00:00:00.000Z",
23    "createdAt": "2023-05-08T12:58:54.407Z",
24    "updatedAt": "2023-05-08T12:58:54.407Z"
25  },
26  {
27    "id": 4,
28    "quantity": 35,
29    "productId": 1,
30    "dateOfSupply": "2022-05-04T00:00:00.000Z",
31    "createdAt": "2023-05-08T12:58:54.407Z",
32    "updatedAt": "2023-05-08T12:58:54.407Z"
33  },
34  {
35    "id": 5,
36    "quantity": 20,
37    "productId": 1
```

Добавление новой поставки

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/v1/supplies/supply`. The 'Body' tab is selected, and the request body is a JSON object: `{ "quantity": 50, "productId": 2, "dateOfSupply": "2022-05-01T00:00:00.000Z" }`. Below the request, the 'Body' tab of the response is selected, showing a JSON object: `{ "id": 13, "quantity": 50, "productId": 2, "dateOfSupply": "2022-05-01T00:00:00.000Z", "updatedAt": "2023-05-08T16:39:50.984Z", "createdAt": "2023-05-08T16:39:50.984Z" }`.

```
POST http://localhost:8000/v1/supplies/supply

{
  "quantity": 50,
  "productId": 2,
  "dateOfSupply": "2022-05-01T00:00:00.000Z"
}
```

```
{
  "id": 13,
  "quantity": 50,
  "productId": 2,
  "dateOfSupply": "2022-05-01T00:00:00.000Z",
  "updatedAt": "2023-05-08T16:39:50.984Z",
  "createdAt": "2023-05-08T16:39:50.984Z"
}
```

Получение поставки по id

The screenshot shows a REST client interface with a GET request to `http://localhost:8000/v1/supplies/supply/13`. The 'Body' tab is selected, and the response body is a JSON object: `{ "id": 13, "quantity": 50, "productId": 2, "dateOfSupply": "2022-05-01T00:00:00.000Z", "createdAt": "2023-05-08T16:39:50.984Z", "updatedAt": "2023-05-08T16:39:50.984Z" }`.

```
GET http://localhost:8000/v1/supplies/supply/13
```

```
{
  "id": 13,
  "quantity": 50,
  "productId": 2,
  "dateOfSupply": "2022-05-01T00:00:00.000Z",
  "createdAt": "2023-05-08T16:39:50.984Z",
  "updatedAt": "2023-05-08T16:39:50.984Z"
}
```


Обновление информации о поставке

POST

http://localhost:8000/v1/supplies/supply

Authorization

Headers (1)

Body

Pre-request Script

form-data

x-www-form-urlencoded

raw

binary

1 {

2 "id": 13,

3 "quantity": 99

4 }

Body

Cookies

Headers (8)

Test Results

Pretty

Raw

Preview

JSON

1 true

Удаление поставки по id

DELETE

http://localhost:8000/v1/supplies/supply/13

Authorization

Headers (1)

Body

Pre-request Script

Tests

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1 {

2 "id": 13,

3 "quantity": 99

4 }

Body

Cookies

Headers (8)

Test Results

Pretty

Raw

Preview

JSON

1 {}

Получение количества поставленных единиц товара за период


GET ▾

http://localhost:8000/v1/supplies/amount?product-id=1

AuthorizationHeaders (1)BodyPre-request ScriptTests

TypeNo Auth ▾

BodyCookiesHeaders (8)Test Results

PrettyRawPreviewJSON ▾

1 ▾ {

2 "amount": 98

3 }|


GET ▾

http://localhost:8000/v1/supplies/amount?product-id=1&date-to=2022-05-02

AuthorizationHeaders (1)BodyPre-request ScriptTests

TypeNo Auth ▾

BodyCookiesHeaders (8)Test Results

PrettyRawPreviewJSON ▾

1 ▾ {

2 "amount": 20

3 }|

Получение списка продаж

GET

http://localhost:8000/v1/sales/sale

Authorization

Headers

Body

Pre-request Script

Tests

Type

No Auth

Body

Cookies

Headers (8)

Test Results

Pretty

Raw

Preview

JSON

```
1 [
2   {
3     "id": 1,
4     "quantity": 2,
5     "price": 15998,
6     "productId": 1,
7     "dateOfSale": "2022-05-02T00:00:00.000Z",
8     "createdAt": "2023-05-08T12:58:54.418Z",
9     "updatedAt": "2023-05-08T12:58:54.418Z"
10  },
11  {
12    "id": 2,
13    "quantity": 1,
14    "price": 7999,
15    "productId": 1,
16    "dateOfSale": "2022-05-03T00:00:00.000Z",
17    "createdAt": "2023-05-08T12:58:54.418Z",
18    "updatedAt": "2023-05-08T12:58:54.418Z"
19  },
20  {
21    "id": 3,
22    "quantity": 4,
23    "price": 31996,
24    "productId": 1,
25    "dateOfSale": "2022-05-04T00:00:00.000Z",
26    "createdAt": "2023-05-08T12:58:54.418Z",
27    "updatedAt": "2023-05-08T12:58:54.418Z"
28  },
29  {
30    "id": 4,
31    "quantity": 10,
32    "price": 79990,
33    "productId": 1,
34    "dateOfSale": "2022-05-05T00:00:00.000Z",
35    "createdAt": "2023-05-08T12:58:54.418Z",
36    "updatedAt": "2023-05-08T12:58:54.418Z"
37  }
38 ]
```

Добавление новой продажи

POST ▾

http://localhost:8000/v1/sales/sale

Authorization

Headers (1)

Body ●

Pre-request Script

Tests

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

JSON (application/json) ▾

1 ▾ {

2 "quantity": 2,

3 "price": 15998,

4 "productId": 1,

5 "dateOfSale": "2022-05-02T00:00:00.000Z"

6 }

Body

Cookies

Headers (8)

Test Results

Pretty

Raw

Preview

JSON ▾



1 ▾ {

2 "id": 10,

3 "quantity": 2,

4 "price": 15998,

5 "productId": 1,

6 "dateOfSale": "2022-05-02T00:00:00.000Z",

7 "updatedAt": "2023-05-08T16:46:37.546Z",

8 "createdAt": "2023-05-08T16:46:37.546Z"

9 }

Добавление новой продажи с актуальной ценой на текущую дату

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/v1/sales`. The request body is a JSON object:

```
{  "quantity": 3,  "productId": 1}
```

The response is displayed in the bottom pane, showing a JSON object with the following fields:

```
{  "id": 11,  "productId": 1,  "quantity": 3,  "price": 23997,  "dateOfSale": "2023-05-08T00:00:00.000Z",  "updatedAt": "2023-05-08T16:47:32.897Z",  "createdAt": "2023-05-08T16:47:32.897Z"}
```

Получение продажи по id

The screenshot shows a REST client interface with a GET request to `http://localhost:8000/v1/sales/sale/11`. The response is displayed in the bottom pane, showing a JSON object with the following fields:

```
{  "id": 11,  "quantity": 3,  "price": 23997,  "productId": 1,  "dateOfSale": "2023-05-08T00:00:00.000Z",  "createdAt": "2023-05-08T16:47:32.897Z",  "updatedAt": "2023-05-08T16:47:32.897Z"}
```

Обновление информации о продаже

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/v1/sales/sale`. The 'Body' tab is selected, and the request body is a JSON object: `{ "id": 11, "price": 11111 }`. The 'Headers' tab shows 8 headers. The 'Test Results' tab shows a single response: `true`.

POST `http://localhost:8000/v1/sales/sale`

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "id": 11,  
3   "price": 11111  
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ▼

```
1 true
```

Удаление продажи по id

The screenshot shows a REST client interface with a DELETE request to `http://localhost:8000/v1/sales/sale/11`. The 'Body' tab is selected, and the request body is a JSON object: `{ "id": 11, "price": 11111 }`. The 'Headers' tab shows 8 headers. The 'Test Results' tab shows a single response: `{}`.

DELETE `http://localhost:8000/v1/sales/sale/11`

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

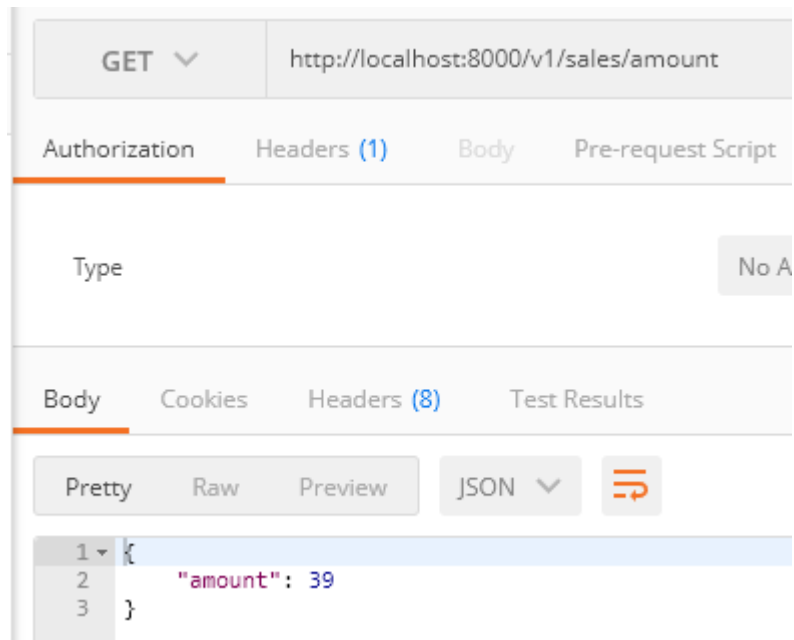
```
1 {  
2   "id": 11,  
3   "price": 11111  
4 }
```

Body Cookies Headers (8) Test Results

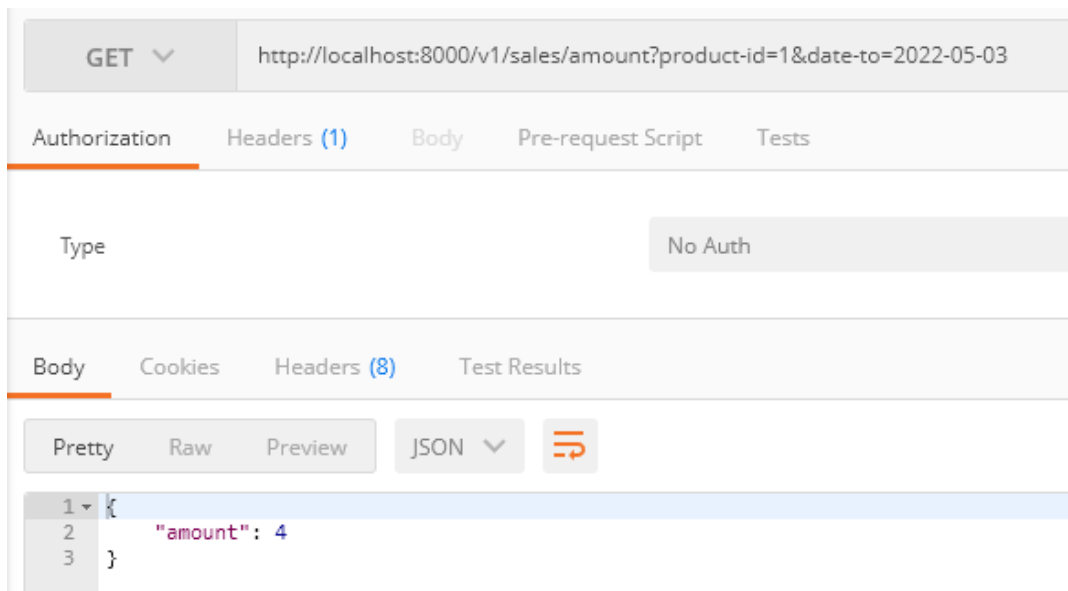
Pretty Raw Preview JSON ▼

```
1 {}
```

Получение суммарного количества проданных единиц всех товаров за все время



Получение суммарного количества проданных единиц товара по id за выбранный период



Получение суммарной выручки от продажи всех товаров за все время

GET ▼ http://localhost:8000/v1/sales/revenue

Authorization Headers (1) Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ▼

```
1 {  
2   "revenue": 227961  
3 }
```

Получение суммарной выручки от продажи товара по id за выбранный период

GET ▼ http://localhost:8000/v1/sales/revenue?product-id=1&date-to=2022-05-03

Authorization Headers (1) Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ▼

```
1 {  
2   "revenue": 31996  
3 }
```


Получение статистики по количеству проданных единиц товара за все время

GET ▾

http://localhost:8000/v1/sales/stat/amount

Authorization

Headers (1)

Body

Pre-request Script

Test Results

Type

No Auth

Body

Cookies

Headers (8)

Test Results

Pretty

Raw

Preview

JSON ▾

≡

1 ▾

2 ▾

3 |

4

5

6 ▾

7

8

9

10 ▾

11

12

13

14 ▾

15

16

17

18 ▾

19

20

21

22 ▾

23

24

25

26

[

{

"dateOfSale": "2022-05-02T00:00:00.000Z",

"amount": 6

},

{

"dateOfSale": "2022-05-03T00:00:00.000Z",

"amount": 6

},

{

"dateOfSale": "2022-05-04T00:00:00.000Z",

"amount": 4

},

{

"dateOfSale": "2022-05-05T00:00:00.000Z",

"amount": 15

},

{

"dateOfSale": "2022-05-06T00:00:00.000Z",

"amount": 5

},

{

"dateOfSale": "2022-05-08T00:00:00.000Z",

"amount": 3

}

]

Получение статистики по количеству проданных единиц товара по id за выбранный период

The screenshot displays a REST client interface. The top bar shows a GET request to the URL `http://localhost:8000/v1/sales/stat/amount?product-id=2&date-to=2022-05-05`. Below this, the 'Headers' tab is active, showing 'No Auth' for the 'Type'. The 'Body' tab is also visible. The response is shown in the 'Test Results' section, formatted as JSON. The response is a list of two objects, each containing 'dateOfSale' and 'amount'.

```
1 [
2   {
3     "dateOfSale": "2022-05-02T00:00:00.000Z",
4     "amount": 2
5   },
6   {
7     "dateOfSale": "2022-05-03T00:00:00.000Z",
8     "amount": 5
9   }
10 ]
```

Получение статистики по выручке от продаж товаров за все время

GET

http://localhost:8000/v1/sales/stat/revenue

Authorization

Headers (1)

Body

Pre-request Script

Tests

Type

No Auth

Body

Cookies

Headers (8)

Test Results

Pretty

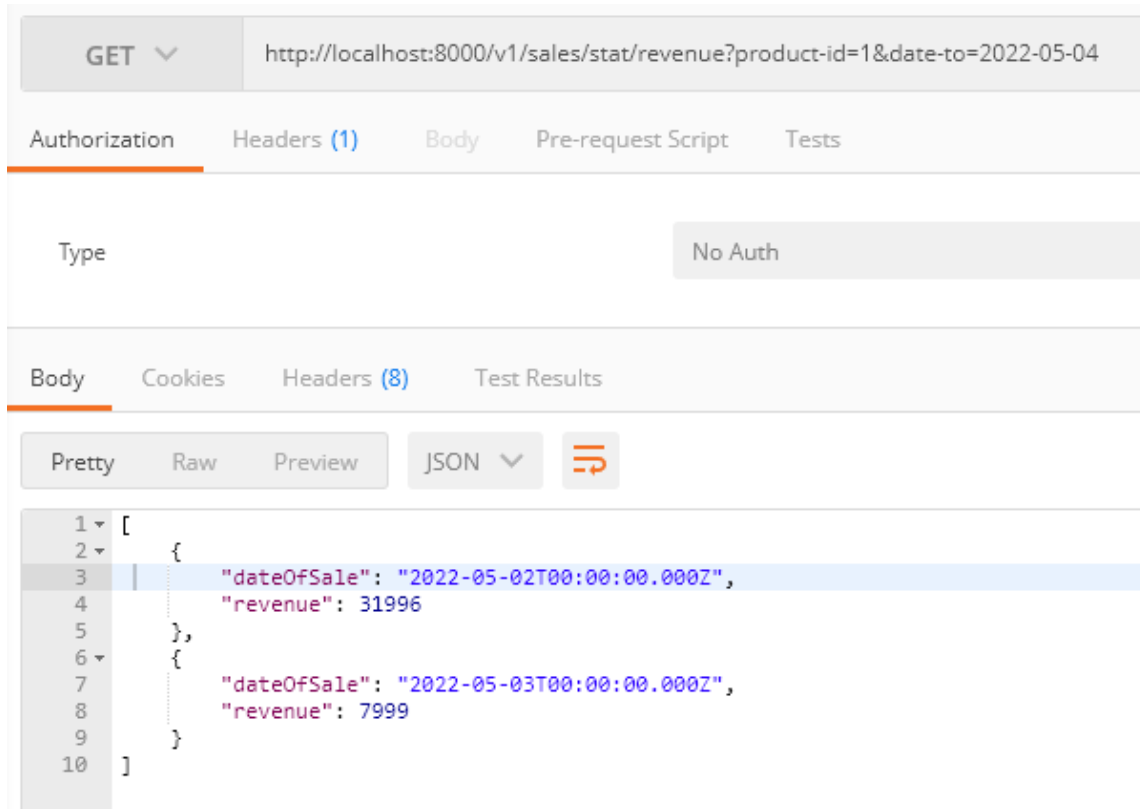
Raw

Preview

JSON

```
1 [
2   {
3     "dateOfSale": "2022-05-02T00:00:00.000Z",
4     "revenue": 36794
5   },
6   {
7     "dateOfSale": "2022-05-03T00:00:00.000Z",
8     "revenue": 19994
9   },
10  {
11    "dateOfSale": "2022-05-04T00:00:00.000Z",
12    "revenue": 31996
13  },
14  {
15    "dateOfSale": "2022-05-05T00:00:00.000Z",
16    "revenue": 91985
17  },
18  {
19    "dateOfSale": "2022-05-06T00:00:00.000Z",
20    "revenue": 39995
21  },
22  {
23    "dateOfSale": "2022-05-08T00:00:00.000Z",
24    "revenue": 7197
25  }
26 ]
```

Получение статистики по выручке от продаж товара по id за выбранный период



Вывод

В ходе пятой практической работы я освоила программу для тестирования API Postman и успешно протестировала все эндпоинты разработанного мной API в рамках ЛР 2.