

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бек-энд разработка

Отчет
Домашняя работа №4

Выполнила: Злотникова Карина Александровна

Группа: К33392

Проверил:
Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

Документирование API из ЛР2 средствами Swagger, Postman

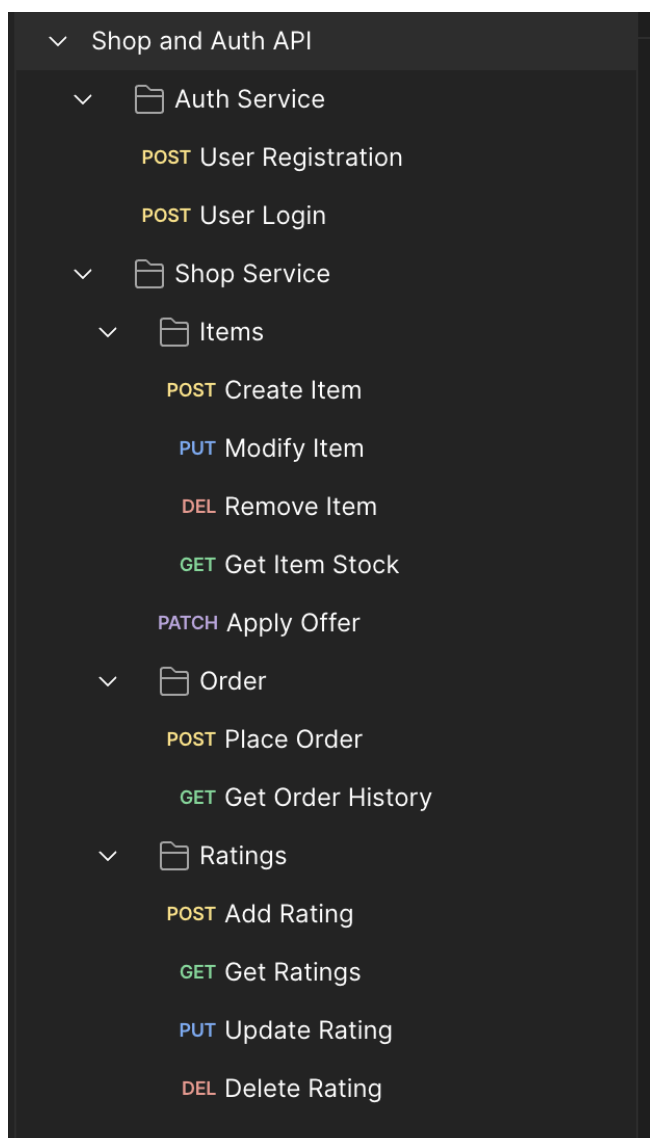
Ход работы

Создадим коллекцию проекта и добавим описание.

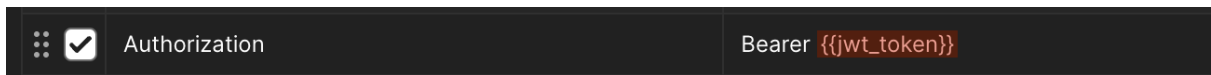
Shop and Auth API

Документация API интернет-магазина

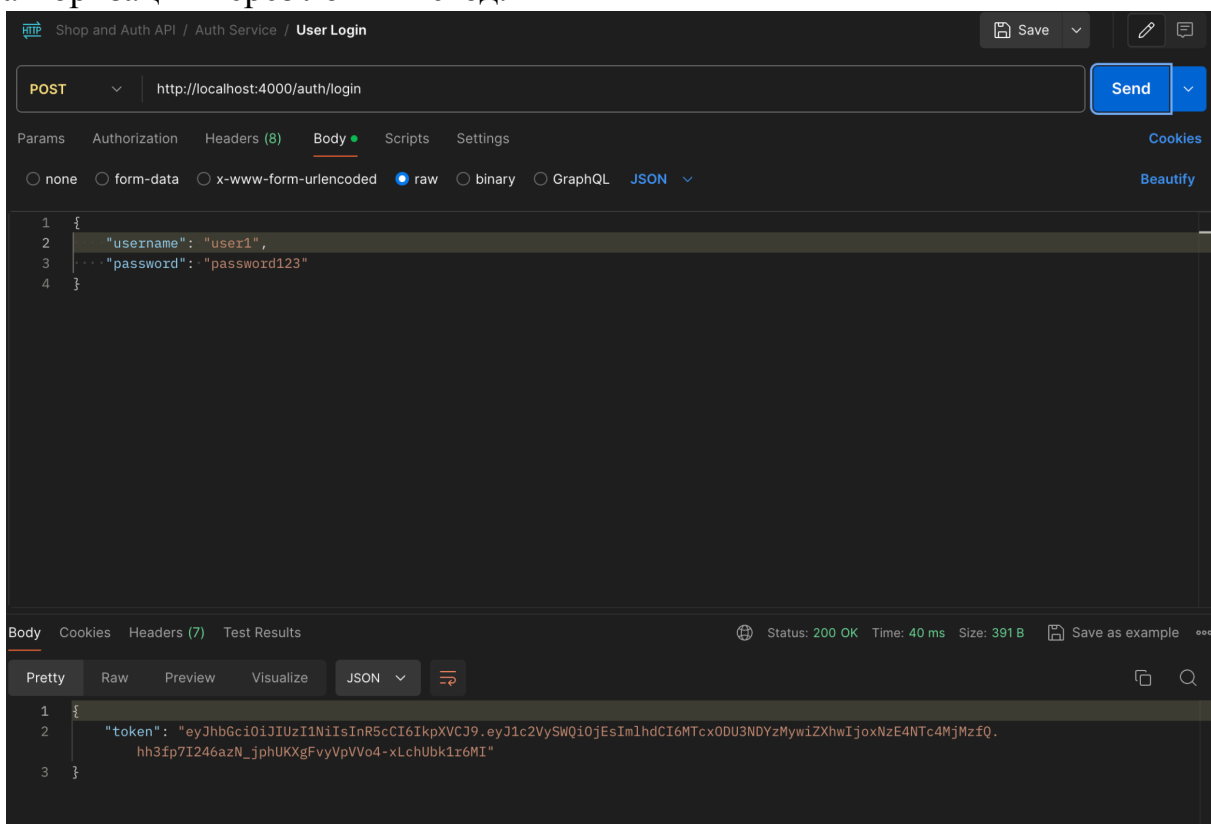
Добавим в Postman все существующие запросы проекта. Сразу разделим всё на удобные подпапки и сервисы.



Для авторизации, в каждом запросе будем указывать в качестве переменной `jwt_token`.



Убедимся что всё работает корректно, создав пользователя и получив токен авторизации через логин метод.



Добавим описания для папок и методов в документацию.

Shop and Auth API

Документация API интернет-магазина

Auth Service

Документация к микросервису авторизаций

POST User Registration

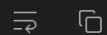
[Open request→](#)

`http://localhost:4000/auth/register`

Позволяет зарегистрировать нового пользователя, вернет ошибку, если пользователь уже существует

Body raw (json)

json



```
{
  "username": "testuser",
  "password": "password123"
}
```

POST Create Item

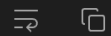
[Open request→](#)

http://localhost:3000/api/items

Позволяет добавить новый товар в магазин

Body raw (json)

json



```
{
  "name": "Test Item",
  "price": 20.50,
  "stock": 100
}
```

PUT Modify Item

[Open request→](#)

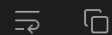
http://localhost:3000/api/items/1

Позволяет изменить товар в магазине по id



Body raw (json)

json



```
{
  "name": "Updated Test Item",
  "price": 25.00,
  "stock": 80
}
```

POST User Login

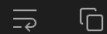
[Open request→](#)

`http://localhost:4000/auth/login`

По username и паролю позволяет получить токен, который в дальнейшем можно использовать для авторизации

Body raw (json)

json



```
{
  "username": "testuser",
  "password": "password123"
}
```

Shop Service

Документация к микросервису магазина

Items

CRUD для работы с предметами в магазине

DELETE Remove Item

[Open request→](#)

`http://localhost:3000/api/items/1`

Позволяет еудалить товар в магазине по id

GET Get Item Stock

[Open request→](#)

`http://localhost:3000/api/items/1/stock`

Позволяет получить количество оставшихся предметов на складе

POST Place Order

[Open request→](#)

http://localhost:3000/api/order

Позволяет произвести заказ

Body raw (json)

json



```
{
  "userId": 1,
  "itemId": 1,
  "quantity": 2
}
```

GET Get Order History

[Open request](#)

http://localhost:3000/api/order/history/1

Позволяет получить историю заказов

POST Add Rating

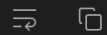
[Open request→](#)

http://localhost:3000/api/ratings

Позволяет оставить оценку на товар

Body raw (json)

json



```
{
  "itemId": 1,
  "rating": 5,
  "comment": "Great product!"
}
```

GET Get Ratings

[Open request→](#)

http://localhost:3000/api/ratings/1

Позволяет получить оценки по id товара

PUT Update Rating

[Open request→](#)

`http://localhost:3000/api/ratings/1`

Позволяет изменить оценку товара от имени пользователя

Body raw (json)

json

```
{
  "rating": 4,
  "comment": "Updated comment"
}
```

DELETE Delete Rating

[Open request→](#)

`http://localhost:3000/api/ratings/1`

Позволяет удалить оценку товара от имени пользователя

Вывод

В ходе работы была я создала документацию в Postman. Postman оказался удобным для документирования и работы с api. Теперь вся информация об api собрана в одном месте.