

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая/Лабораторная работа

Выполнил:

Пластун Елизавета Олеговна

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

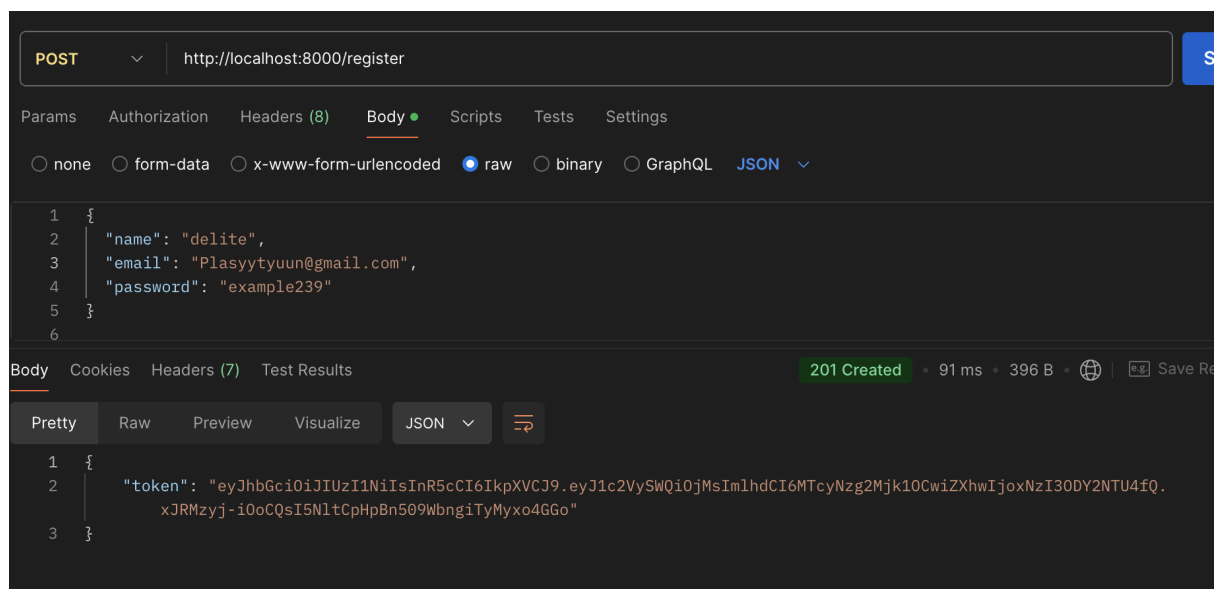
2022 г.

Задача

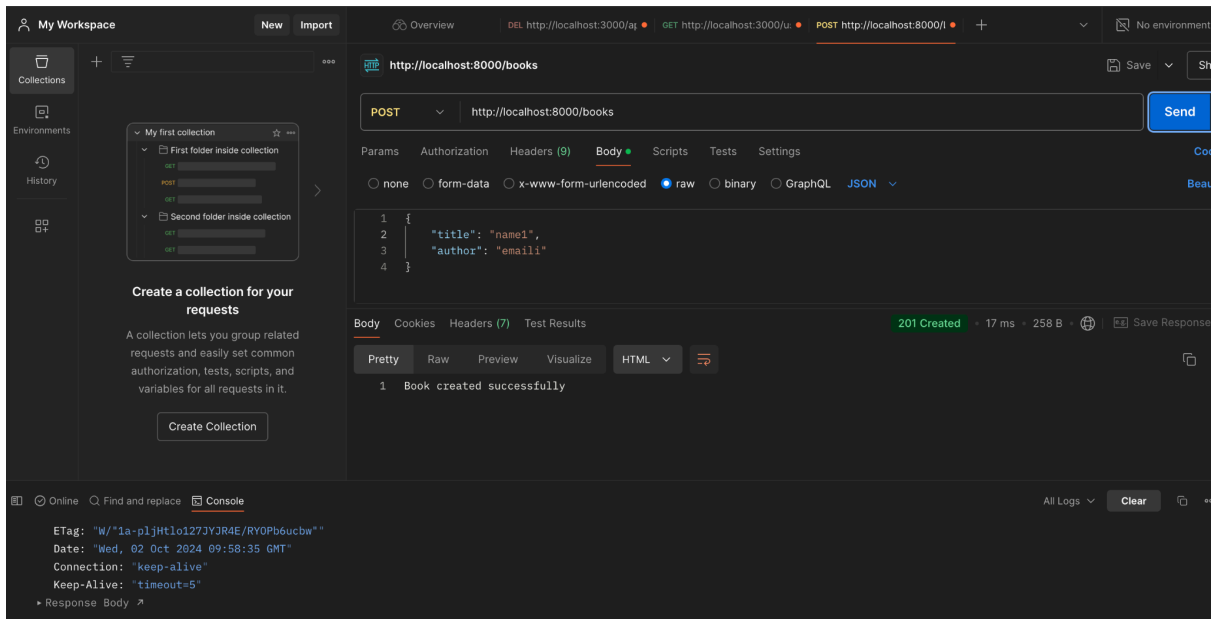
По выбранному варианту необходимо будет реализовать RESTful API средствами express + typescript (используя ранее написанный boilerplate). Вариант: сервис для буккроссинга. Требуемый функционал: регистрация, авторизация, создание профиля, создание списка своих книг, создание заявок на обмен книгами, работа с заявками на обмен.

Ход работы

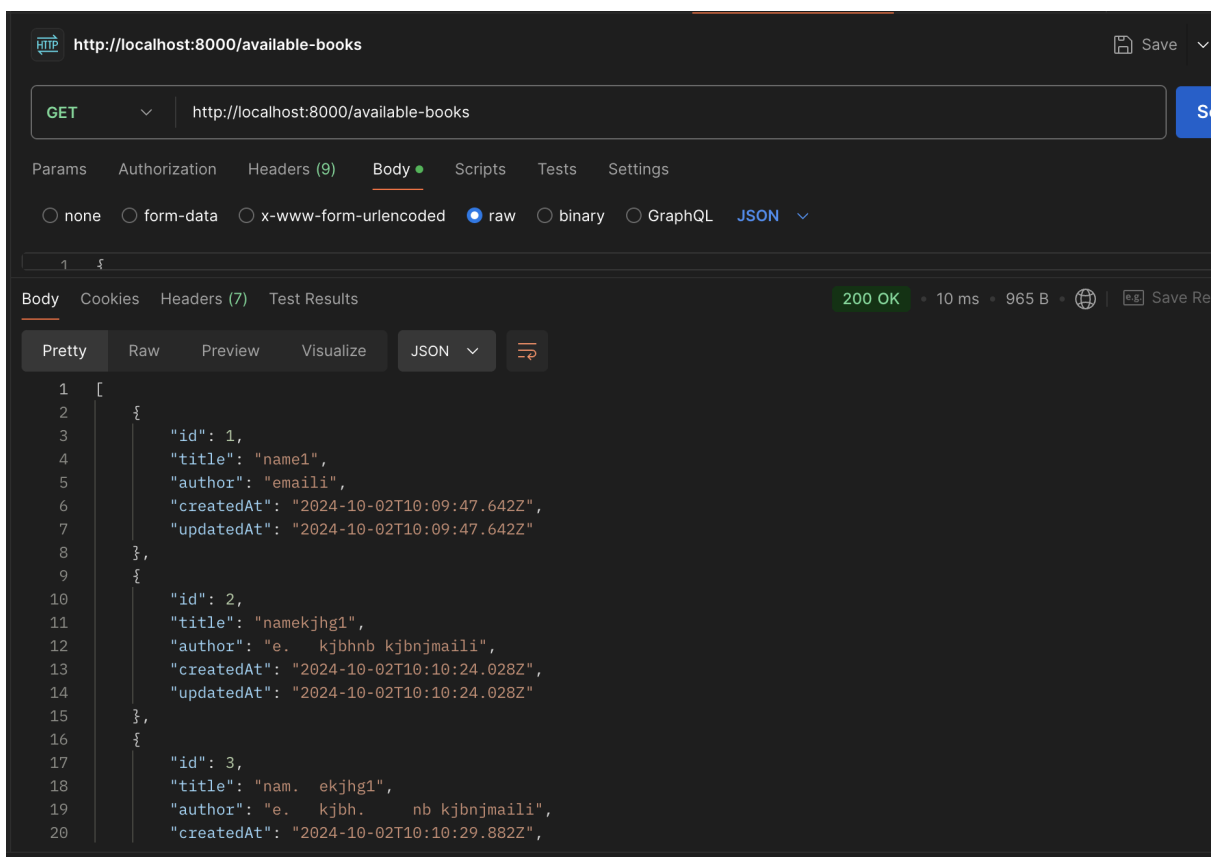
Для начала создадим модели для книги, также таблицу связывающую юзера с книгой, запроса на обмен, профиля. Далее реализовываем регистрацию с паролем, который хэшируется и авторизацию. Регистрация



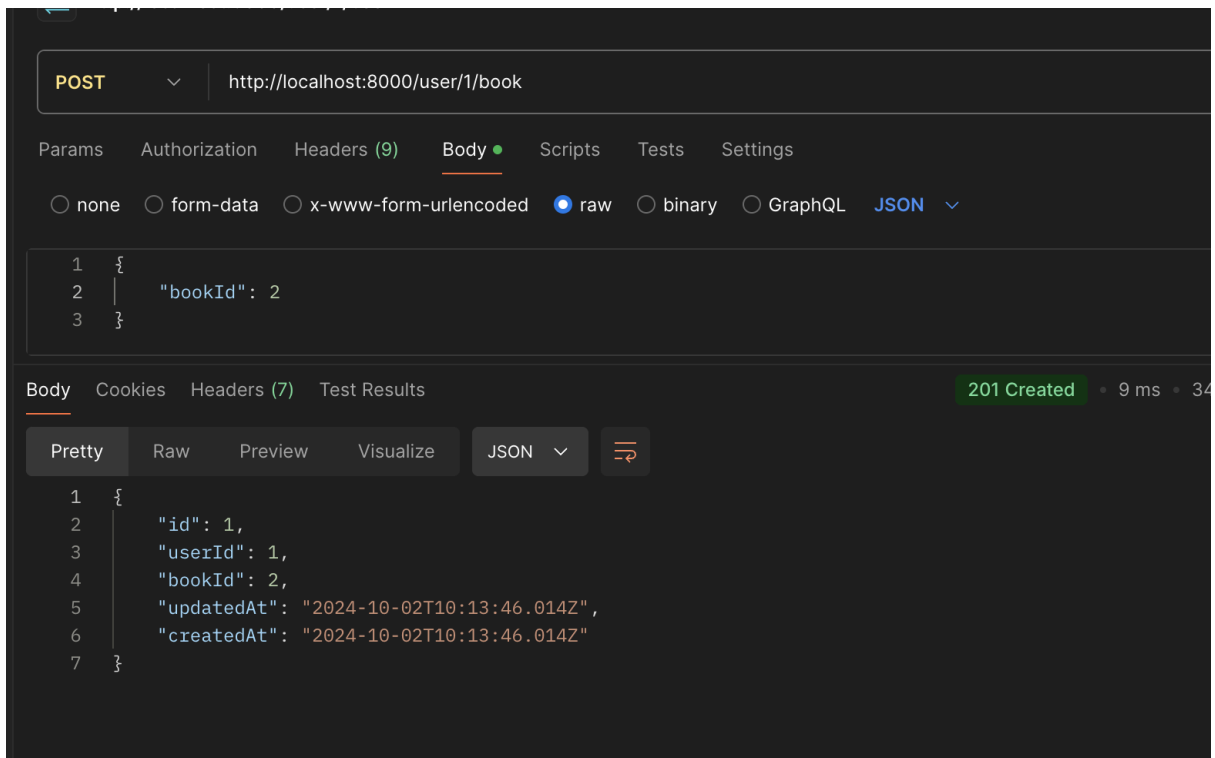
Далее создаем функцию добавления книги в список пользователя. Сначала потенциальный админ загружает определенный список книг, который можно просматривать, и из которого пользователя могут добавлять себе в список доступные книги. Создание книги



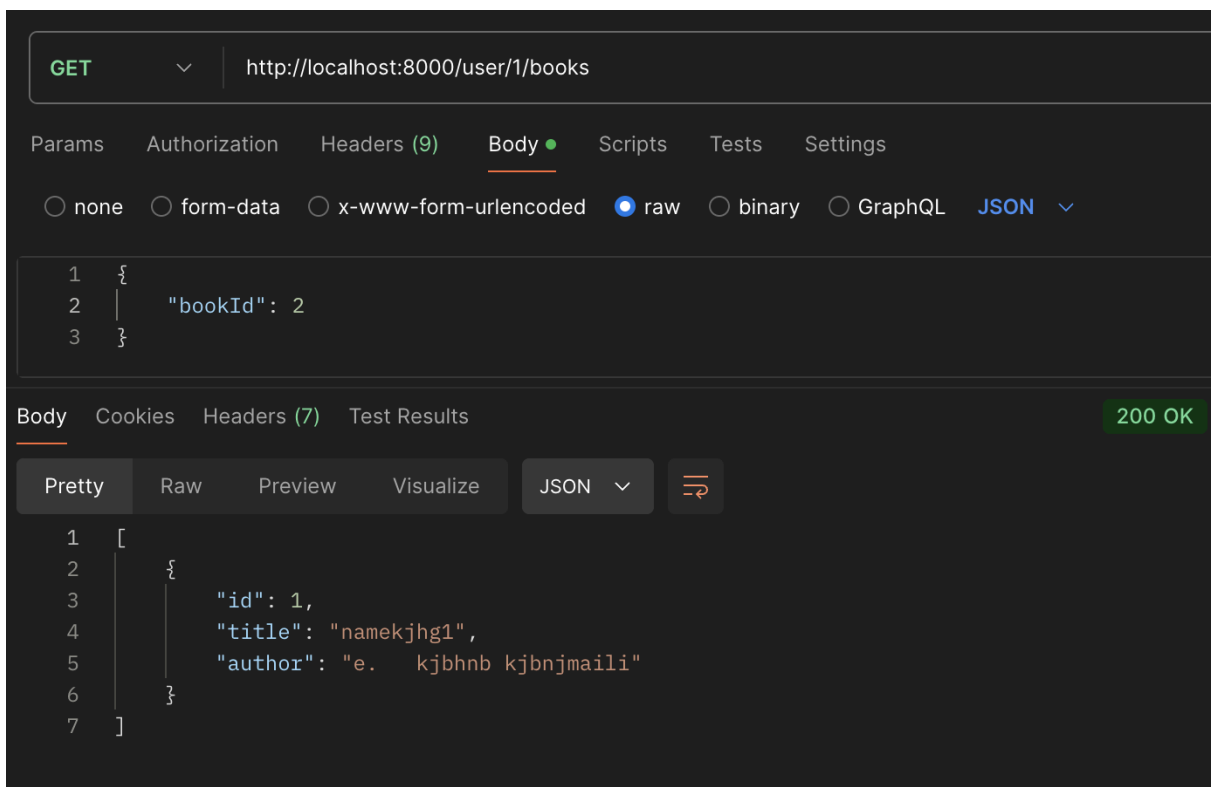
Получения списка доступных книг



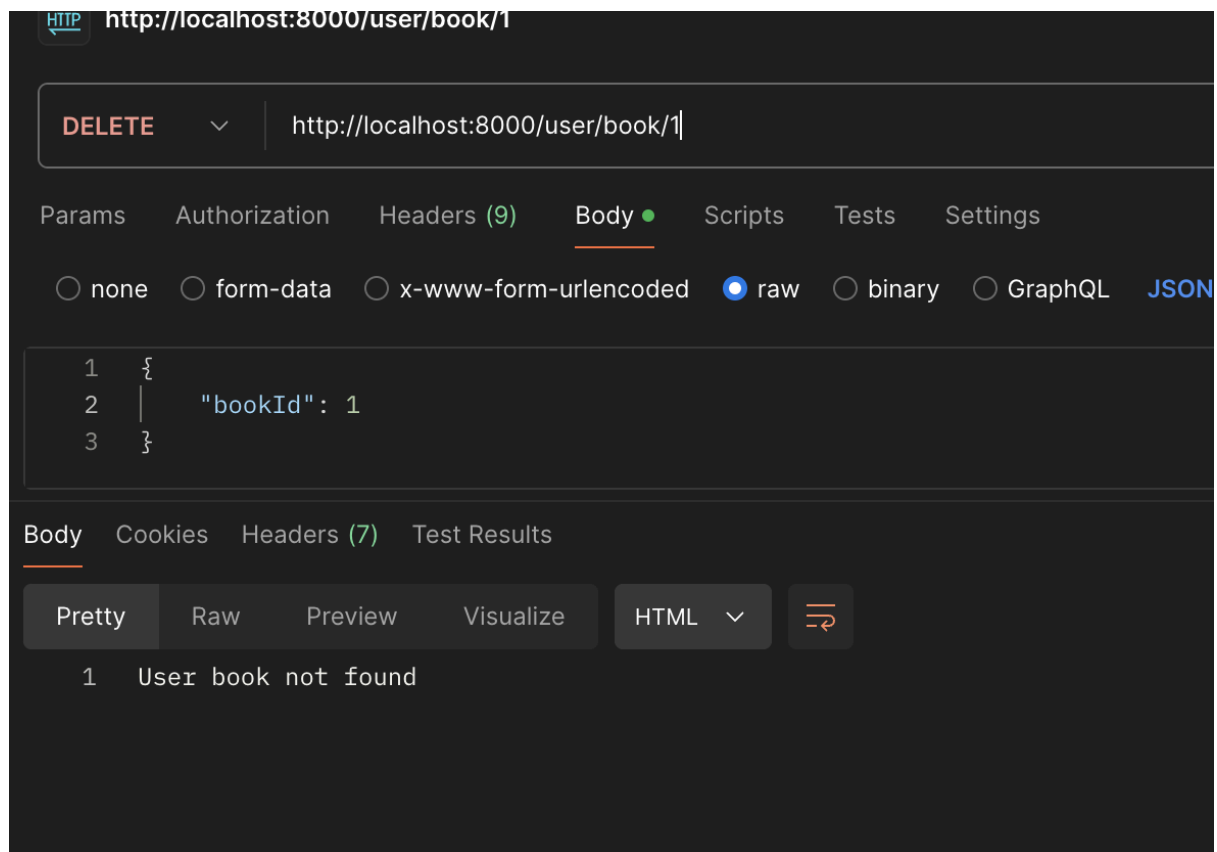
Добавление новой книги в список пользователя



Получение списка книг пользователя



Удаление книги из списка пользователя



При регистрации у пользователя создается пустой профиль, для которого была создана функция его редактирования, чтобы заполнить поля bio и location. Создание профиля или его редактирование

HTTP **http://localhost:8000/users/2/profile**

POST **http://localhost:8000/users/2/profile**

Params Authorization Headers (9) **Body** Scripts Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL [JSON](#)

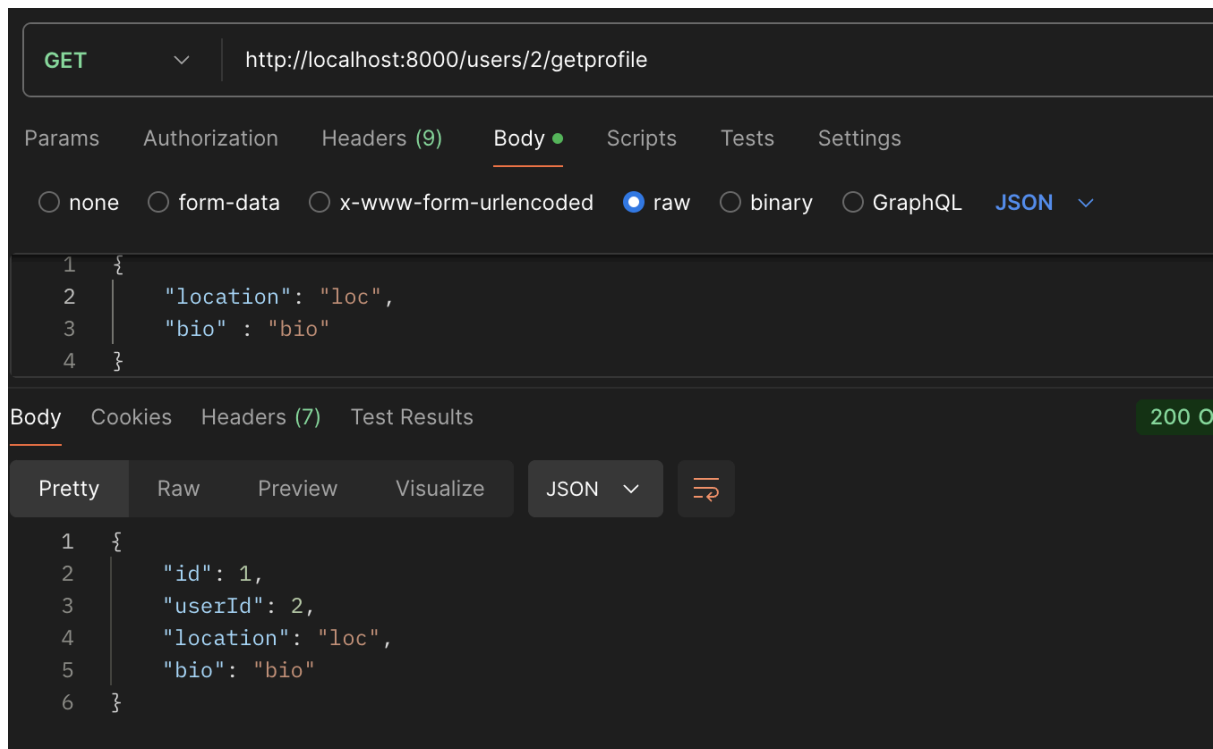
```
1 {
2   "location": "loc",
3   "bio" : "bio"
4 }
```

Body Cookies Headers (7) Test Results

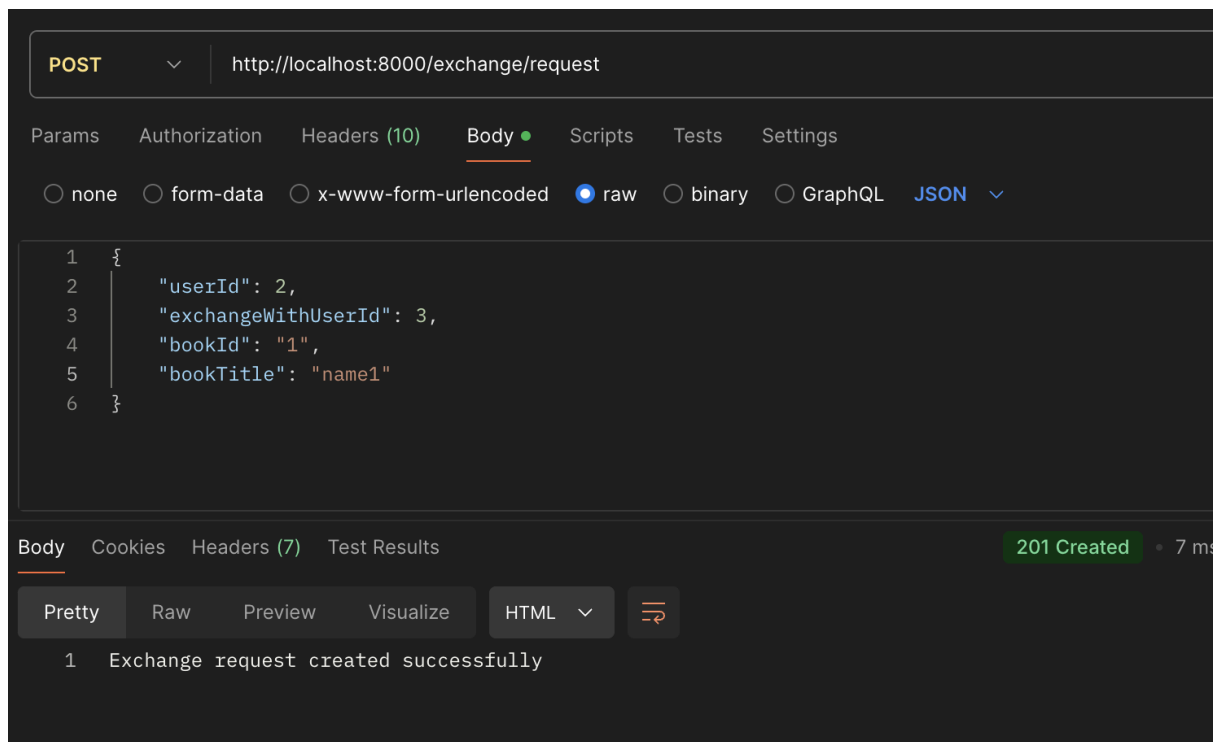
Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "userId": 2,
4   "location": "loc",
5   "bio": "bio"
6 }
```

Получение профиля пользователя по id



Далее создаем функционал для обмена книгами. Пользователь может создать запрос, удалить его. Также можно получить список созданных заявок у конкретного пользователя по id. Создание запроса на обмен



Получение списка заявок по id пользователя

GET http://localhost:8000/exchange/user/2

Params Authorization Headers (10) Body Scripts Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

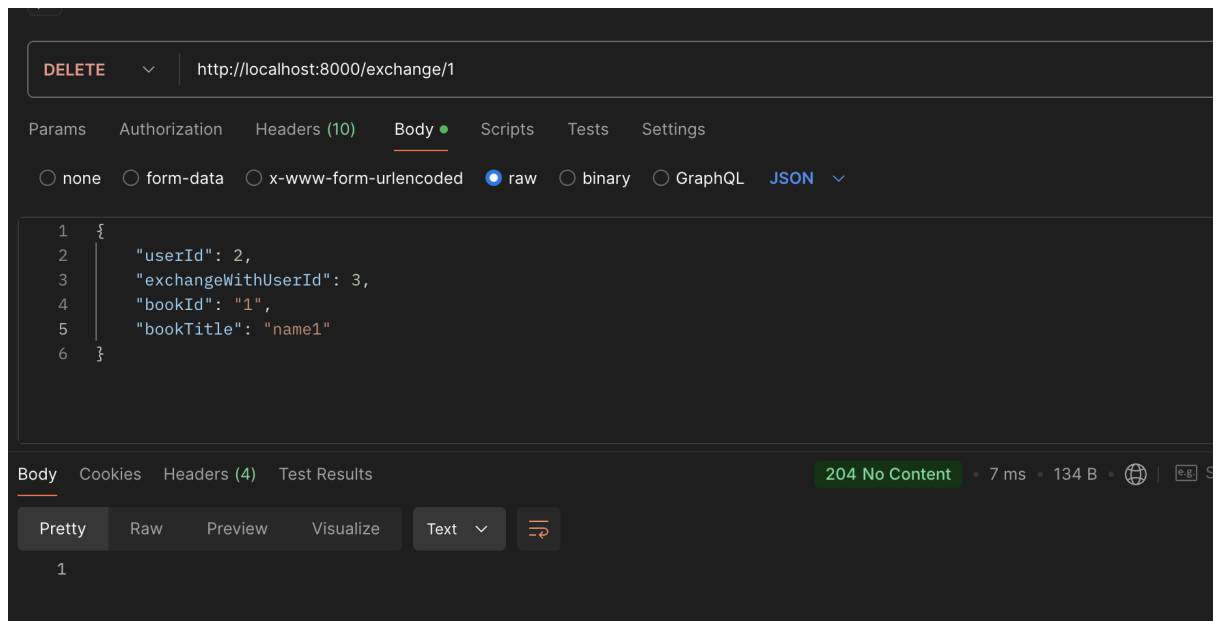
```
1 {
2   "userId": 2,
3   "exchangeWithUserId": 3,
4   "bookId": "1",
5   "bookTitle": "name1"
6 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "userId": 2,
4   "exchangeWithUserId": 3,
5   "bookId": 1,
6   "bookTitle": "name1",
7   "status": "pending",
8   "createdAt": "2024-10-02T10:31:10.366Z",
9   "updatedAt": "2024-10-02T10:31:10.366Z"
10 }
11
12 }
```

Удаление заявки



Вывод

Во второй лабораторной работе удалось создать собственный сервис для букросинга на основе ранее написанного boilerplate с использованием express + typescript. Реализованы следующие функции: регистрация и авторизация, создание запросов на обмен книгами и работа с ними, возможность создания и редактирование профиля, создание книг и добавление их в личный список пользователя.