

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Практическая работа 5

Выполнила:

Герулайтите Габриэля

К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

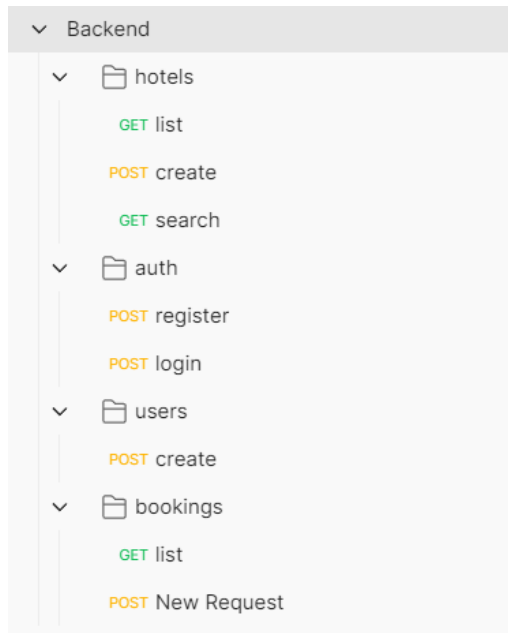
2022 г.

# Задача

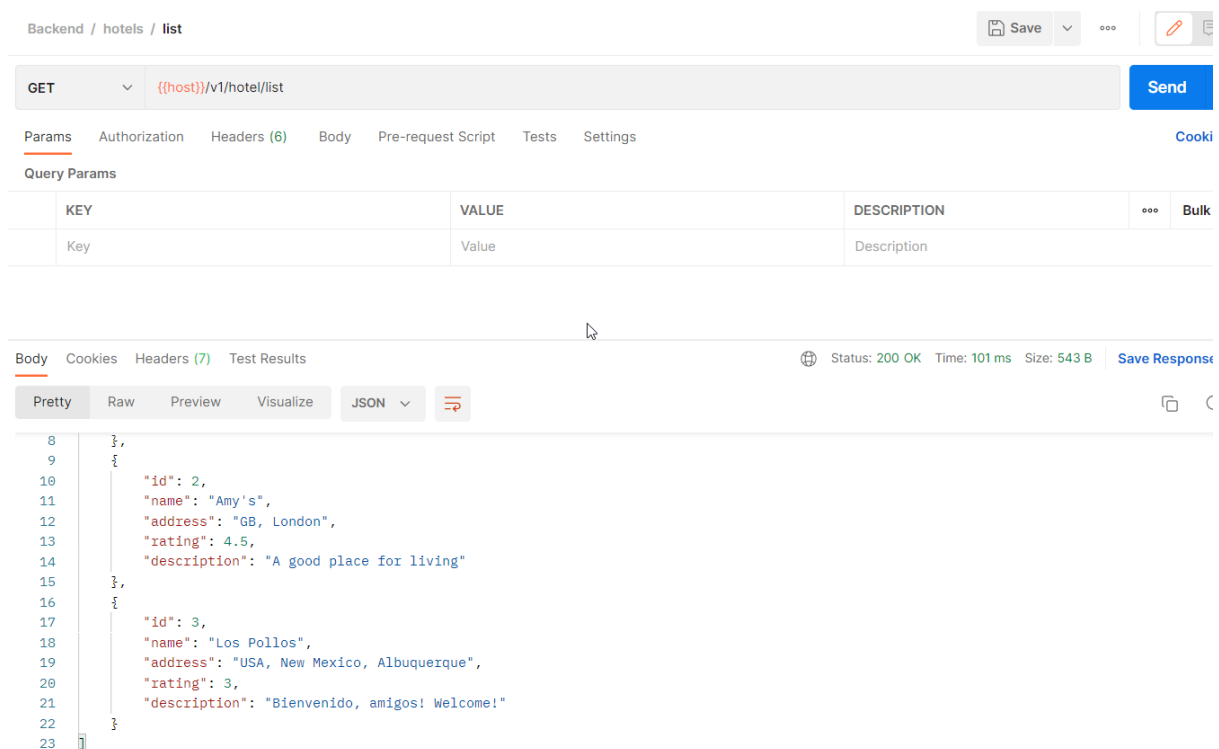
Тестирование API из ЛР2 средствами Postman

## Ход работы

Структура запросов:



Получение списка отелей:



## Создание карточки отеля:

Backend / hotels / create

POST ▼ `{{host}}/v1/hotel/create`

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

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

```
1  {
2    "name": "Los Pollos",
3    "address": "USA, New Mexico, Albuquerque",
4    "rating": 3.0,
5    "description": "Bienvenido, amigos! Welcome!"
6  }
7  }
8
```

Body Cookies Headers (7) Test Results ⌕ Stat

Pretty Raw Preview Visualize **JSON** ▼ ≡

```
1  {
2    "record": {
3      "name": "Los Pollos",
4      "address": "USA, New Mexico, Albuquerque",
5      "rating": 3,
6      "description": "Bienvenido, amigos! Welcome!",
7      "id": 4
8    },
9    "msg": "Successfully create hotel"
10 }
```

## Поиск отеля по адресу:

Backend / hotels / search

GET ▼ `{{host}}/v1/hotel/list?q=US`

Params **●** Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	q	US
	Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize **JSON** ▼ ≡

```
1  {
2    "id": 3,
3    "name": "Los Pollos",
4    "address": "USA, New Mexico, Albuquerque",
5    "rating": 3,
6    "description": "Bienvenido, amigos! Welcome!"
7  },
8 }
```

## Авторизация. Создание пользователя/регистрация

Backend / auth / register

POST `{{host}}/v1/auth/register`

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

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

```
1 {
2   "firstName": "Elly",
3   "lastName": "Haay",
4   "email": "ellyhay@gmail.com",
5   "password": "password2"
6 }
7 }
```

Body Cookies Headers (7) Test Results Status: 201 Created

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "firstName": "Elly",
3   "lastName": "Haay",
4   "email": "ellyhay@gmail.com",
5   "password": "password2",
6   "id": 2
7 }
```

## Вход

Backend / auth / login

POST `{{host}}/v1/auth/login`

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

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

```
1 {
2   "email": "ellyhay@gmail.com",
3   "password": "password2"
4 }
```

Body Cookies Headers (7) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6Im1waWF0IjoxNjU0MjE4fQ.dCJAGHD_fpgneX"
3 }
```

Выдаётся токен, который, благодаря скрипту, автоматически подставляется в бронирования `{{accessToken}}`, и сессия поддерживается некоторое время:

Backend / auth / login

POST ⌵ {{host}}/v1/auth/login

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

```
1 const responseJson = pm.response.json();
2
3 pm.globals.set('accessToken', responseJson.accessToken)
4
```

Backend / bookings / New Request

Save ⌵

POST ⌵ {{host}}/v1/booking/create

Params Authorization ● Headers (9) Body ● Pre-request Script Tests Settings

Type

Bearer Token ⌵

The authorization header will be automatically generated when you send the request.  
[Learn more about authorization](#)

⚠ Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)

Token

{{accessToken}}

## Бронирование

Backend / bookings / New Request

POST ⌵ {{host}}/v1/booking/create

Params Authorization ● Headers (9) Body ● Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ⌵

```
1 {
2   "hotel": 3
3 }
4
```

Body Cookies Headers (7) Test Results

Status: 200 OK

Pretty

Raw

Preview

Visualize

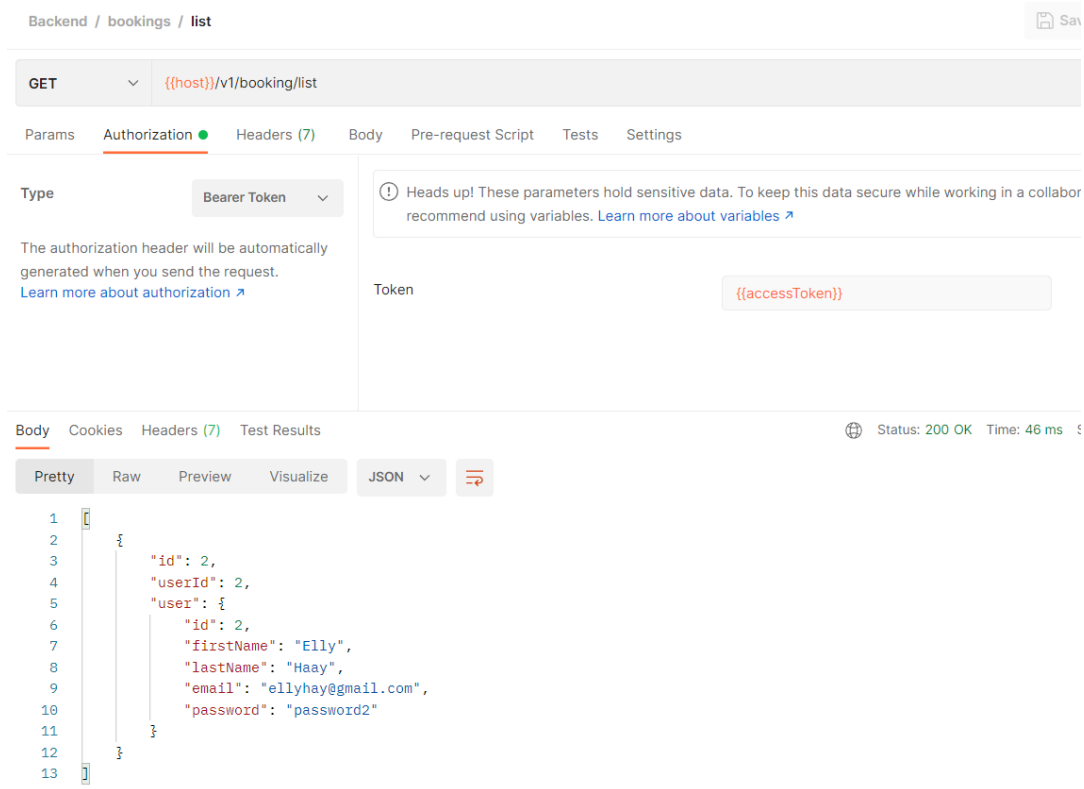
JSON ⌵



```
1 {
2   "record": {
3     "hotel": 3,
4     "userId": 2,
5     "id": 2
6   },
7   "msg": "Successfully attend this event"
8 }
```

Видим, что пользователь с id2 (Elly Наау), успешно создала бронирование отеля с id3 и это бронирование с id2.

Это ещё возможно проверить, посмотрев список бронирований пользователя:



## Вывод

В ходе работы ознакомилась с программой Postman, с помощью которой был протестирован API (сервис бронирований отелей) и собрана коллекция запросов.