

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа 2

Выполнил:

Тюмин Никита

Группа K33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

1. Продумать свою собственную модель пользователя
2. Реализовать набор из CRUD-методов для работы с пользователями средствами Express + Sequelize
3. Написать запрос для получения пользователя по id/email

Ход работы

1. Инициализируем модуль, устанавливаем необходимые зависимости, создаем модель пользователя, создаем миграции, запускаем миграции и запускаем сервер разработки:

```
npm init
```

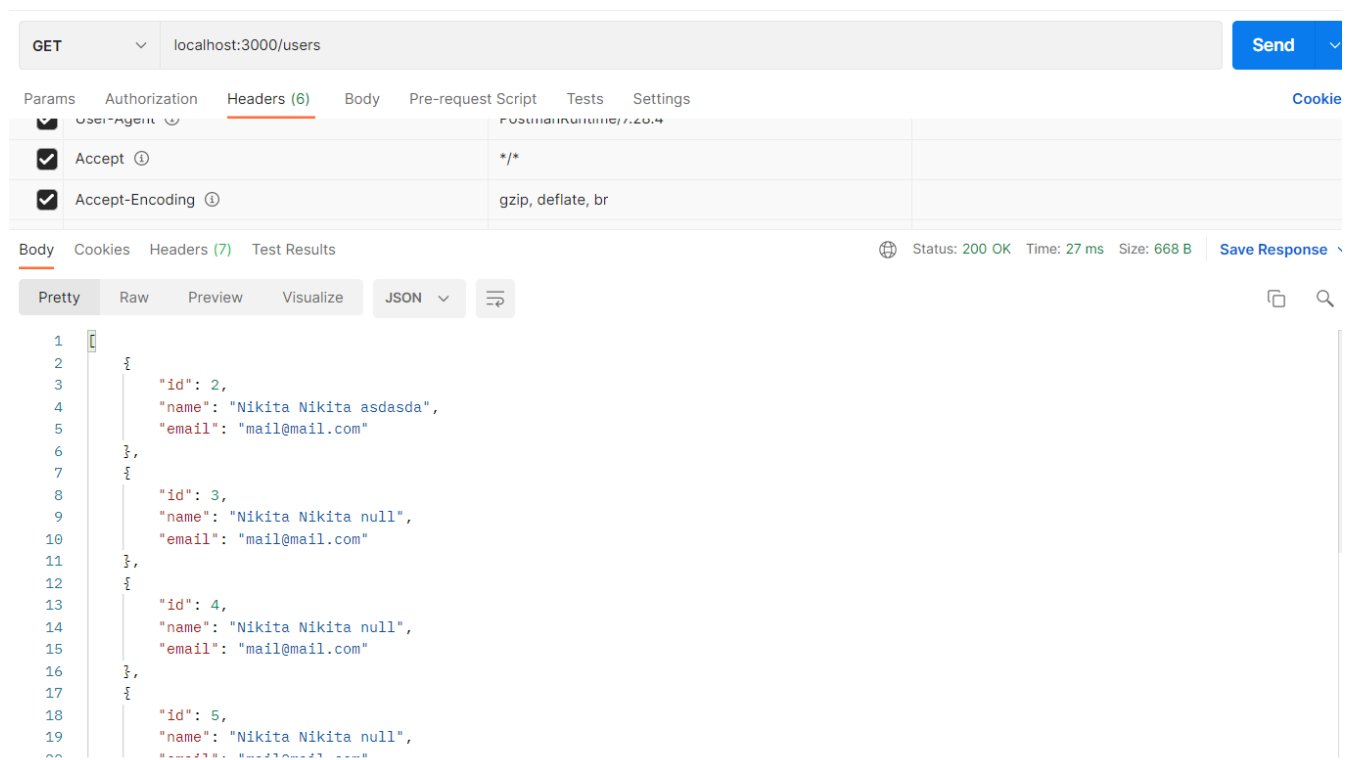
```
npm i express sequelize sqlite3 sequelize-cli express-validator bcrypt
```

```
npm sequelize init
```

```
sequelize db:migrate
```

```
node index.js
```

2. Листинг пользователей:



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:3000/users
- Status:** 200 OK
- Time:** 27 ms
- Size:** 668 B

The response body is displayed in JSON format, showing an array of 5 user objects:

```
1 {
2   {
3     "id": 2,
4     "name": "Nikita Nikita asdasda",
5     "email": "mail@mail.com"
6   },
7   {
8     "id": 3,
9     "name": "Nikita Nikita null",
10    "email": "mail@mail.com"
11  },
12  {
13    "id": 4,
14    "name": "Nikita Nikita null",
15    "email": "mail@mail.com"
16  },
17  {
18    "id": 5,
19    "name": "Nikita Nikita null",
20    "email": "mail@mail.com"
21  },
22  {
23    "id": 6,
24    "name": "Nikita Nikita null",
25    "email": "mail@mail.com"
26  }
27 }
```

3. Получение пользователя:

GETlocalhost:3000/users/2

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookie

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 16 ms

Size: 371 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "id": 2,
3    "firstName": "Nikita",
4    "lastName": "Nikita",
5    "patronymic": "asdasda",
6    "email": "mail@mail.com",
7    "createdAt": "2024-03-13T12:27:00.000Z"
8  }
```

Bootcamp

Runner

Trash

4. Создание пользователя:

POSTlocalhost:3000/users

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookie

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1  {
2    "first_name": "Nikita",
3    "last_name": "Nikita",
4    "patronymic": null,
5    "email": "mail@mail.com",
6    "password": "123qwe123"
7  }
```

Body

Cookies

Headers (7)

Test Results

Status: 201 Created

Time: 90 ms

Size: 438 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "fullName": "Nikita Nikita null",
3    "id": 9,
4    "firstName": "Nikita",
5    "lastName": "Nikita",
6    "patronymic": null,
7    "email": "mail@mail.com",
8    "password": "$2b$10$LB6meca1yu281eVN3495e.3001V2TTQ8A3D8SD4gbup9mivcI3Ciu"
9  }
```

Bootcamp

Runner

Trash

5. Редактирование пользователя:

PUTlocalhost:3000/users/2

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

```
1 {
2   ... "first_name": "Nikita",
3   ... "last_name": "Nikita",
4   ... "patronymic": "asdasda"
5 }
```

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 10 msSize: 371 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 2,
3   "firstName": "Nikita",
4   "lastName": "Nikita",
5   "patronymic": "asdasda",
6   "email": "mail@mail.com",
7   "createdAt": "2024-03-13T12:27:00.000Z"
8 }
```

BootcampRunnerTrash

6. Удаление пользователя:

DELETElocalhost:3000/users/2

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

Query Params

KEY	VALUE	DESCRIPTION	Bulk Ed
Key	Value	Description	

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 20 msSize: 273 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "detail": "User deleted successfully"
3 }
```

BootcampRunnerTrash

Вывод

В данной домашней работе было реализовано небольшое приложения, которое обрабатывает запросы для CRUD-операций над пользователем при помощи библиотеки `express` и `sequelize` для работы с базой данных.