

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"

Факультет «Инфокоммуникационных технологий»

Направление подготовки «09.03.03 Прикладная информатика»

Бакалаврская программа «Мобильные и сетевые технологии»

ОТЧЕТ
по домашнему заданию 2
по дисциплине «Бэк-энд разработка»

Выполнила: Матрохина А.С., К33402

Проверил: Добряков Д.И.

Дата: 30.03.2022

Санкт-Петербург
2022

Ход выполнения работы:

1) Создана модель пользователя, включающая в себя следующие поля:

- a. Имя
- b. Фамилия
- c. Отчество
- d. Email
- e. Биография

```
User.init({
  firstName: DataTypes.STRING,
  lastName: DataTypes.STRING,
  middleName: DataTypes.STRING,
  email: DataTypes.STRING,
  bio: DataTypes.STRING
}, {
  sequelize,
  modelName: 'User',
});
```

2) Реализованы запросы CRUD методов с помощью фреймворка Express и Sequelize.

- a. Get /users – список всех пользователей
- b. Post /users – создание пользователя
- c. Get /users/:id – пользователь по id
- d. Put /users/:id – редактирование пользователя по id
- e. Delete /users/:id - удаление пользователя по id

The screenshot displays a REST client interface for a POST request to `http://localhost:8080/users`. The request body is a JSON object with the following fields: `firstName` (Ann), `lastName` (Matrokhina), `middleName` (Sergeevna), `email` (email@email.ru), and `bio` (Hi! I'm Ann!). The response status is 200 OK, and the response body is a JSON object that includes all the request fields plus an `id` of 1, an `updatedAt` timestamp, and a `createdAt` timestamp.

```
{
  "id": 1,
  "firstName": "Ann",
  "lastName": "Matrokhina",
  "middleName": "Sergeevna",
  "email": "email@email.ru",
  "bio": "Hi! I'm Ann!",
  "updatedAt": "2022-03-29T23:06:56.166Z",
  "createdAt": "2022-03-29T23:06:56.166Z"
}
```

PUT http://localhost:8080/

No Environment

http://localhost:8080/users/1

Save

PUT

http://localhost:8080/users/1

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Beautiful

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "bio": "This is my new bio!!"
3 }
```

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 24 ms

Size: 266 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "msg": "user has been updated"
3 }
```

GET http://localhost:8080/

No Environment

http://localhost:8080/users/1

Save

GET

http://localhost:8080/users/1

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Beautiful

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "bio": "This is my new bio!!"
3 }
```

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 11 ms

Size: 443 B

Save Response

Pretty

Raw

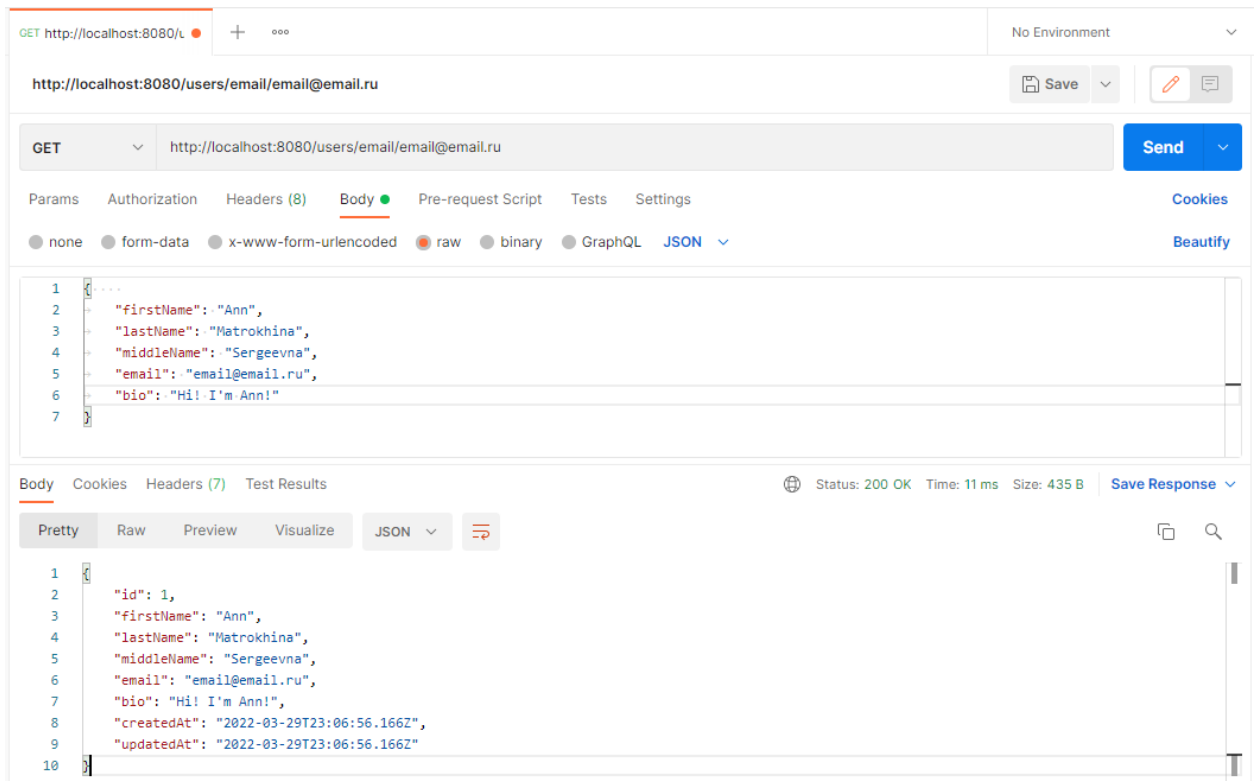
Preview

Visualize

JSON

```
1 {
2   "id": 1,
3   "firstName": "Ann",
4   "lastName": "Matrokhina",
5   "middleName": "Sergeevna",
6   "email": "email@email.ru",
7   "bio": "This is my new bio!!",
8   "createdAt": "2022-03-29T23:06:56.166Z",
9   "updatedAt": "2022-03-29T23:08:51.023Z"
10 }
```

3) Реализован метод получения пользователя по email



Вывод:

Был освоен фреймворк Express и ORM Sequelize, основы работы с моделями, создание CRUD-методов и фильтров.