

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа

Выполнил:

Саунин Антон

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

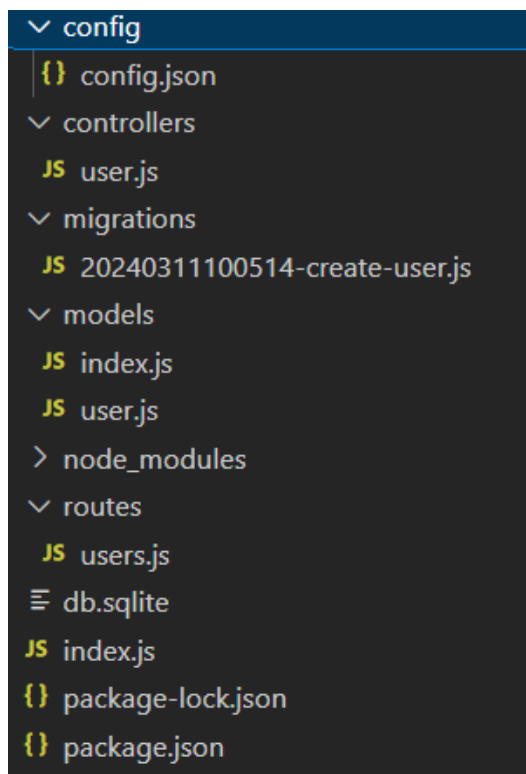
2024 г.

Задача

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

Ход работы

Структура папок:



Запуск сервера и определение маршрута для пользователей

```
var express = require("express");
var usersRouter = require("./routes/users");

var app = express();
app.use(express.json());
const port = 3000;
app.use("/users", usersRouter);

app.listen(port, () => console.log(`listen on port ${port}`));
```

Определение маршрутов к методам с пользователями

```
router.post("/", UserController.createUser);
router.get("/", UserController.getAllUsers);
router.get("/:id", UserController.getUserById);
router.put("/:id", UserController.updateUser);
router.delete("/:id", UserController.deleteUser);
router.get("/email/:email", UserController.getUserByEmail);

module.exports = router;
```

Создания пользователя

```
createUser: async (req, res) => {
  try {
    const newUser = await User.create(req.body);
    res.status(201).json(newUser);
  } catch (error) {
    res.status(400).json({ message: error.message });
  }
},
```

The screenshot shows a REST client interface with a POST request to `http://localhost:3000/users`. The request body is a JSON object with the following fields: `username` ("anton"), `email` ("anton2@test.com"), `password` ("12345678"), and `phone` ("88005553535"). The response is a 201 Created status with a JSON object containing the created user's details, including an `id` of 4, `updatedAt`, and `createdAt` timestamps.

POST `http://localhost:3000/users` Send

Params Authorization Headers (8) Body Scripts Settings Cookies

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

```
1 {
2   "username": "anton",
3   "email": "anton2@test.com",
4   "password": "12345678",
5   "phone": "88005553535"
6 }
```

Body Cookies Headers (7) Test Results 201 Created • 54 ms • 416 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 4,
3   "username": "anton",
4   "email": "anton2@test.com",
5   "password": "12345678",
6   "phone": "88005553535",
7   "updatedAt": "2024-09-17T22:43:47.583Z",
8   "createdAt": "2024-09-17T22:43:47.583Z"
9 }
```

Получение пользователей

```
getAllUsers: async (req, res) => {
  try {
    const users = await User.findAll();
    res.json(users);
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
},

getUserById: async (req, res) => {
  const { id } = req.params;
  try {
    const user = await User.findByPk(id);
    if (!user) {
      return res.status(404).json({ message: "User not found" });
    }
    res.json(user);
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
},
```

GET http://localhost:3000/users/

Params Authorization Headers (6) Body Scripts Settings

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

This request does not have a body

Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1  [
2    {
3      "id": 3,
4      "username": "anton",
5      "email": "anton@test.com",
6      "password": "12345678",
7      "phone": "88005553535",
8      "createdAt": "2024-09-09 19:07:51.297 +00:00",
9      "updatedAt": "2024-09-09 19:07:51.297 +00:00"
10   },
11   {
12     "id": 4,
13     "username": "anton",
14     "email": "anton2@test.com",
15     "password": "12345678",
16     "phone": "88005553535",
17     "createdAt": "2024-09-17 22:43:47.583 +00:00",
18     "updatedAt": "2024-09-17 22:43:47.583 +00:00"
19   }
20 ]
```

GET http://localhost:3000/users/4

Params Authorization Headers (6) Body Scripts Settings

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

This request does not have a body

Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1  {
2    "id": 4,
3    "username": "anton",
4    "email": "anton2@test.com",
5    "password": "12345678",
6    "phone": "88005553535",
7    "createdAt": "2024-09-17 22:43:47.583 +00:00",
8    "updatedAt": "2024-09-17 22:43:47.583 +00:00"
9  }
```

Удаление пользователя

```
deleteUser: async (req, res) => {
  const { id } = req.params;
  try {
    const deletedRowCount = await User.destroy({ where: { id } });
    if (deletedRowCount === 0) {
      return res.json({ message: "User not found" });
    }
    res.json({ message: "User deleted" });
  } catch (error) {
    res.json({ message: error.message });
  }
},
```

DELETE

▼

http://localhost:3000/users/8

ParamsAuthorizationHeaders (8)Body ●ScriptsSe

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐

1

{

2

"username": "Ivan",

3

"email": "anton25@test.com",

4

"password": "12345678",

5

"phone": "87864456546"

6

}

bodyCookiesHeaders (7)Test Results

Pretty

Raw

Preview

Visualize

JSON ▼

↺

1

{

2

"message": "User deleted"

3

}

Поиск пользователя по email:

GET

▼

http://localhost:3000/users/email/anton25@test.com

ParamsAuthorizationHeaders (6)BodyScriptsSettings

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

This request does not have a body

bodyCookiesHeaders (7)Test Results

PrettyRawPreviewVisualizeJSON▼

```
1  {
2    "id": 6,
3    "username": "Ivan",
4    "email": "anton25@test.com",
5    "password": "12345678",
6    "phone": "87864456546",
7    "createdAt": "2024-09-17 23:10:47.143 +00:00",
8    "updatedAt": "2024-09-17 23:10:47.143 +00:00"
9  }
```

Вывод

В ходе данной работы был реализован набор из CRUD-методов для работы с пользователями средствами Express + Sequelize.