

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Практическая работа 2

Выполнил:

Старовойтова Елизавета

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

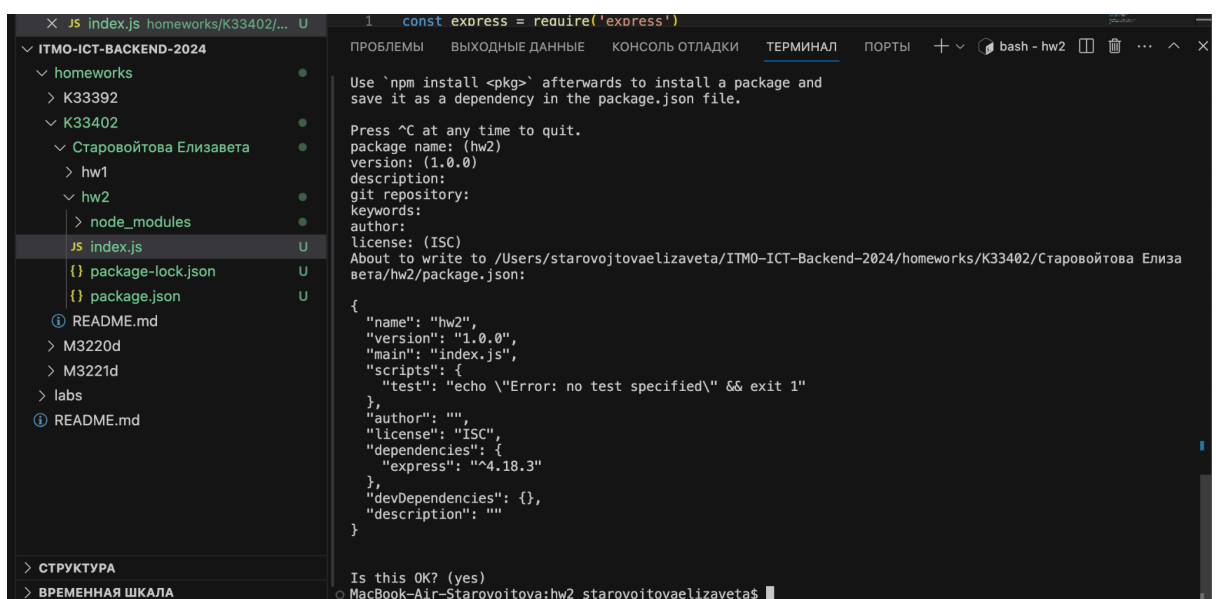
2024 г.

## 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

### 1. Инициализация npm пакета.



```
1 const express = require('express')
```

Use `npm install <pkg>` afterwards to install a package and save it as a dependency in the package.json file.

Press ^C at any time to quit.

package name: (hw2)

version: (1.0.0)

description:

git repository:

keywords:

author:

license: (ISC)

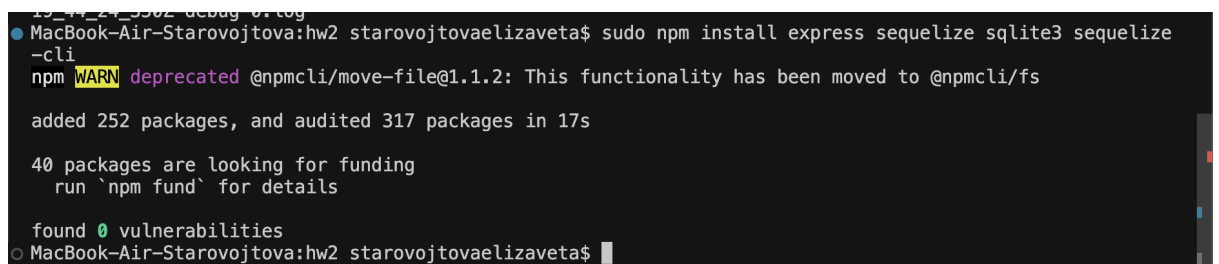
About to write to /Users/starovojtovaelizaveta/ITMO-ICT-Backend-2024/homeworks/K33402/Старовойтова Елизавета/hw2/package.json:

```
{
  "name": "hw2",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.18.3"
  },
  "devDependencies": {},
  "description": ""
}
```

Is this OK? (yes)

MacBook-Air-Starovojtova:hw2 starovojtovaelizaveta\$

### 2. Установка.



```
MacBook-Air-Starovojtova:hw2 starovojtovaelizaveta$ sudo npm install express sequelize sqlite3 sequelize-cli
npm WARN deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs

added 252 packages, and audited 317 packages in 17s

40 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
MacBook-Air-Starovojtova:hw2 starovojtovaelizaveta$
```

### 3. Инициализация sequelize-cli.

```
Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.1]

Created "config/config.json"
Successfully created models folder at "/Users/starovojtovaelizaveta/ITMO-ICT-Backend-2024/homeworks/K33402/Старовойтова Елизавета/hw2/models".
Successfully created migrations folder at "/Users/starovojtovaelizaveta/ITMO-ICT-Backend-2024/homeworks/K33402/Старовойтова Елизавета/hw2/migrations".
Successfully created seeders folder at "/Users/starovojtovaelizaveta/ITMO-ICT-Backend-2024/homeworks/K33402/Старовойтова Елизавета/hw2/seeders".
MacBook-Air-Starovojtova:hw2 starovojtovaelizaveta$
```

#### 4. Генерация модели с помощью cli.

```
MacBook-Air-Starovojtova:hw2 starovojtovaelizaveta$ npx sequelize-cli model:generate --name User --attributes "username:string, email:string, password:string, firstName:string, lastName:string, isAdmin:boolean"

Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.1]

New model was created at /Users/starovojtovaelizaveta/ITMO-ICT-Backend-2024/homeworks/K33402/Старовойтова Елизавета/hw2/models/user.js .
New migration was created at /Users/starovojtovaelizaveta/ITMO-ICT-Backend-2024/homeworks/K33402/Старовойтова Елизавета/hw2/migrations/20240319200111-create-user.js .
MacBook-Air-Starovojtova:hw2 starovojtovaelizaveta$
MacBook-Air-Starovojtova:hw2 starovojtovaelizaveta$
```

```
npx sequelize-cli model:generate --name User --attributes
"username:string, email:string, password:string, firstName:string,
lastName:string, isAdmin:boolean"
```

#### 5. Миграция.

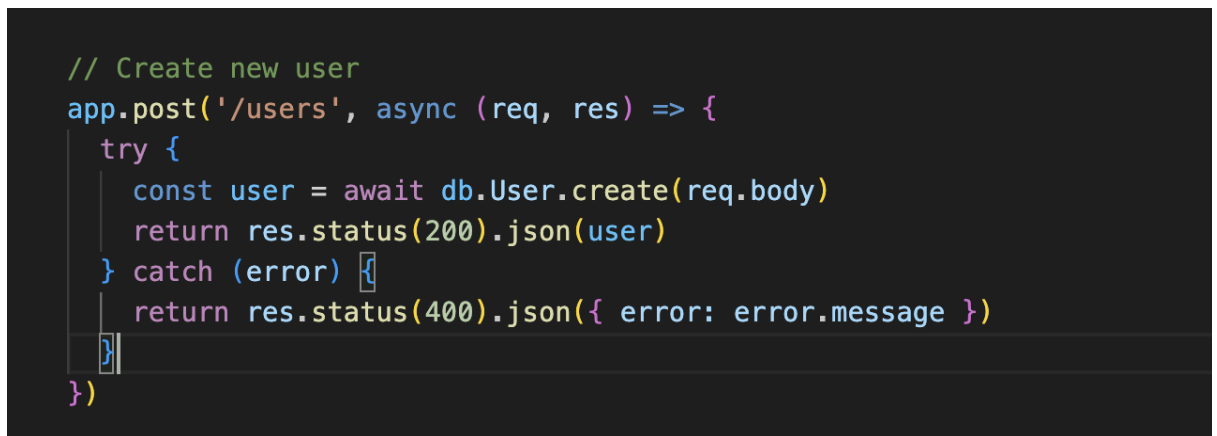
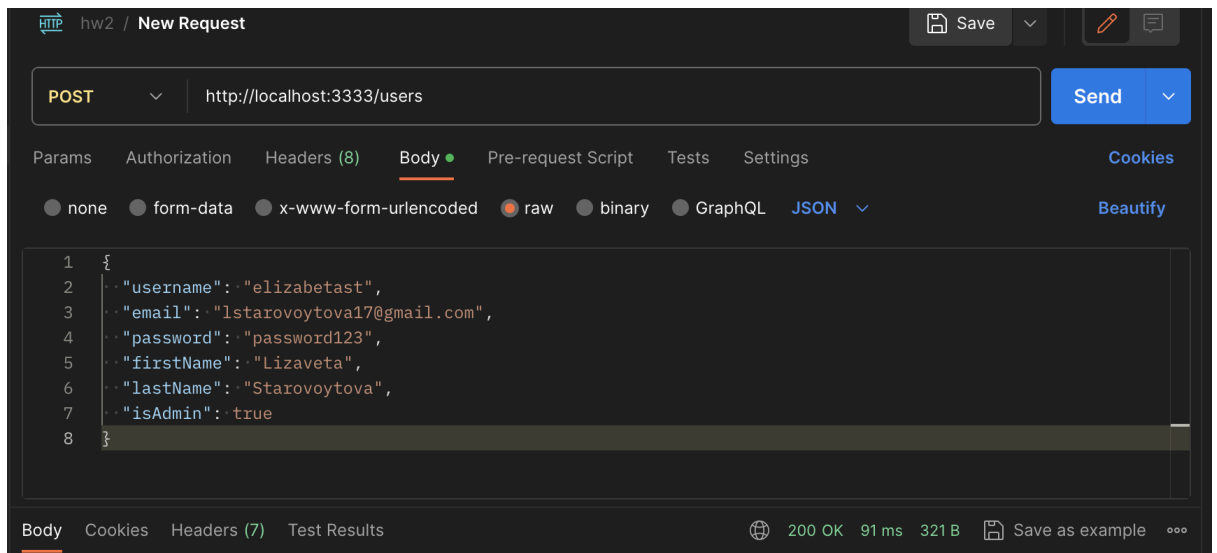
```
MacBook-Air-Starovojtova:hw2 starovojtovaelizaveta$ npx sequelize-cli db:migrate

Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.1]

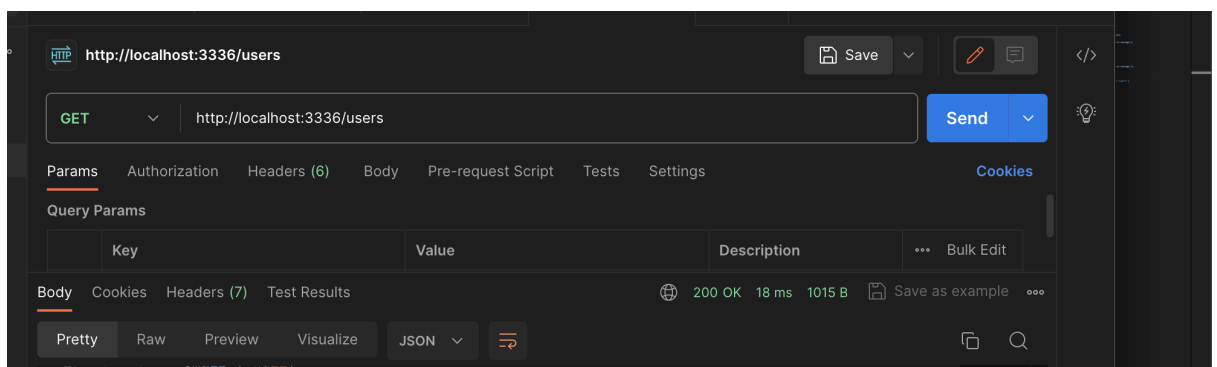
Loaded configuration file "config/config.json".
Using environment "development".
== 20240319200111-create-user: migrating =====
== 20240319200111-create-user: migrated (0.014s)

MacBook-Air-Starovojtova:hw2 starovojtovaelizaveta$
```

#### 6. Create

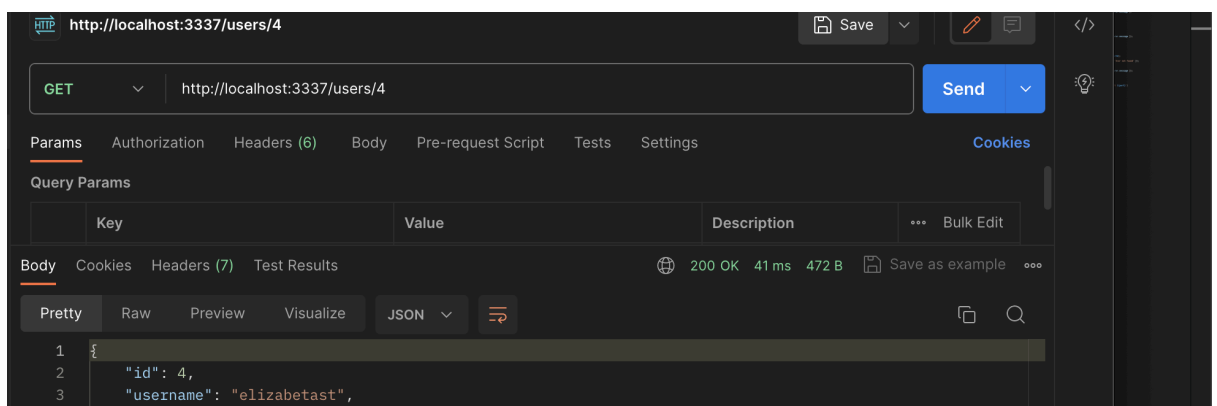


Get



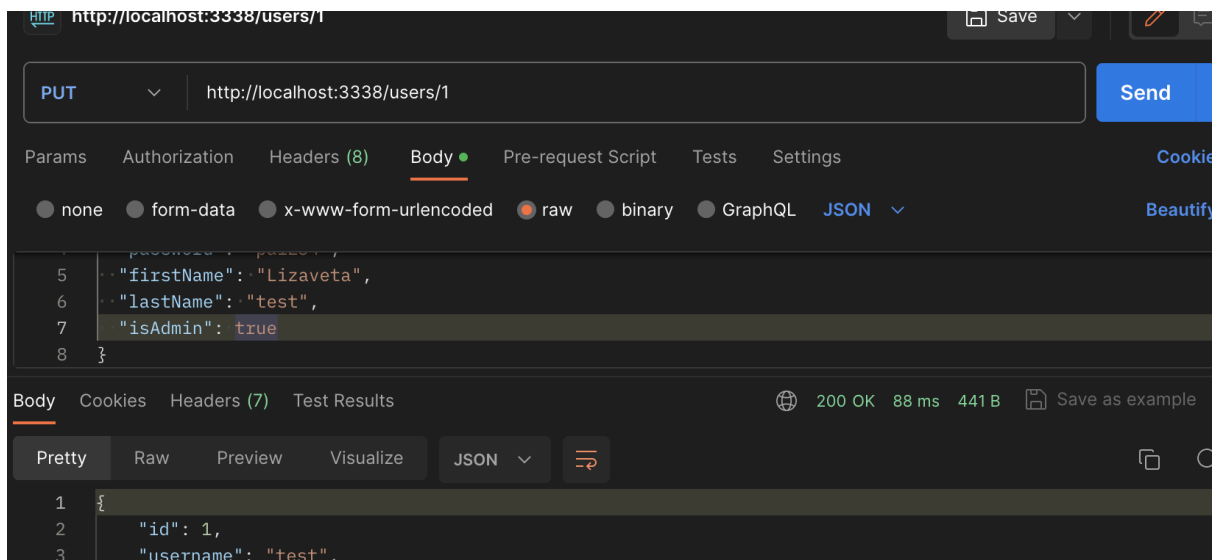
```
// Get users
app.get('/users', async (req, res) => {
  try {
    const users = await db.User.findAll();
    return res.json(users);
  } catch (error) {
    return res.status(500).json({ error: error.message });
  }
});
```

## Get by ID



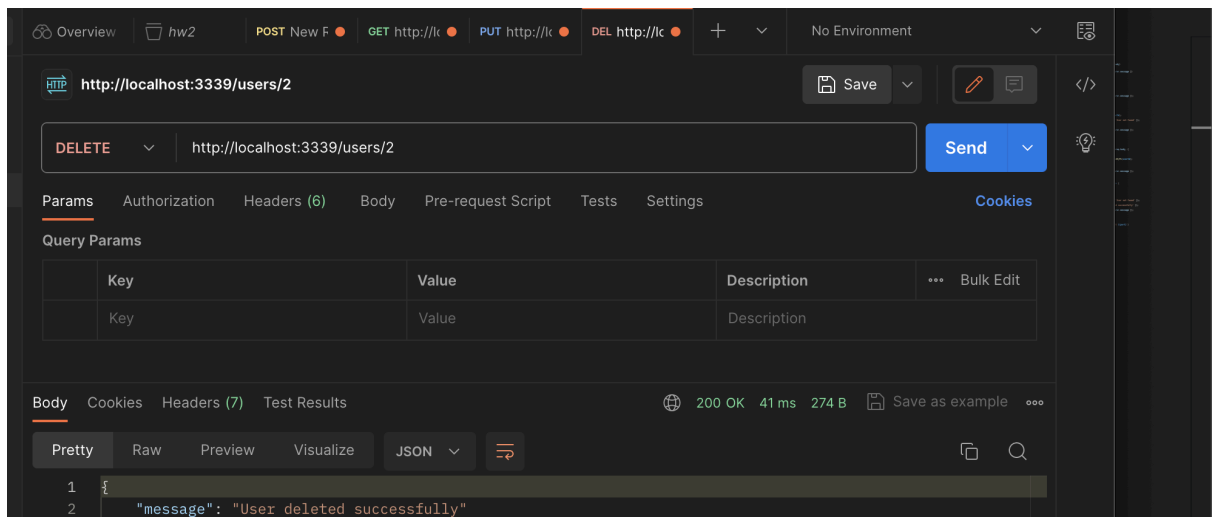
```
// Get user by ID
app.get('/users/:id', async (req, res) => {
  try {
    const userId = req.params.id;
    const user = await db.User.findByPk(userId);
    if (!user) {
      return res.status(404).json({ error: 'User not found' });
    }
    return res.json(user);
  } catch (error) {
    return res.status(500).json({ error: error.message });
  }
});
```

## Update



```
// Update user
app.put('/users/:id', async (req, res) => {
  try {
    const userId = req.params.id;
    const [updated] = await db.User.update(req.body, {
      where: { id: userId }
    });
    if (updated) {
      const updatedUser = await db.User.findByPk(userId);
      return res.json(updatedUser);
    }
    throw new Error('User not found');
  } catch (error) {
    return res.status(404).json({ error: error.message });
  }
});
```

Delete



```
// Delete user
app.delete('/users/:id', async (req, res) => {
  try {
    const userId = req.params.id;
    const deleted = await db.User.destroy({
      where: { id: userId }
    });
    if (!deleted) {
      return res.status(404).json({ error: 'User not found' });
    }
    return res.json({ message: 'User deleted successfully' });
  } catch (error) {
    return res.status(500).json({ error: error.message });
  }
});
```

## Вывод

В данной практической работе я познакомилась ORM Sequelize, командной утилитой и микрофреймворком Express. Создала модель пользователя и реализовала CRUD-методы.