

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ О ДОМАШНЕЙ РАБОТЕ № 2
по теме: Знакомство с ORM Sequelize
по дисциплине: Бэк-энд разработка

Специальность:

09.03.03 Мобильные и сетевые технологии

Выполнила:

Малютина Е.А., К33402

Проверил:

Добряков Д.И.

Санкт-Петербург,

2024

Задание.

- Придумать 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.

Были установлены библиотека Sequelize и драйвер для PostgreSQL (Рисунок 1).

```
(base) ekaterinamalutina@Air-Ekaterina-4 Backend % npm install --save sequelize  
  
added 23 packages, and audited 24 packages in 3s  
  
1 package is looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
(base) ekaterinamalutina@Air-Ekaterina-4 Backend % npm install --save pg pg-hstore  
  
added 17 packages, and audited 41 packages in 3s  
  
1 package is looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities
```

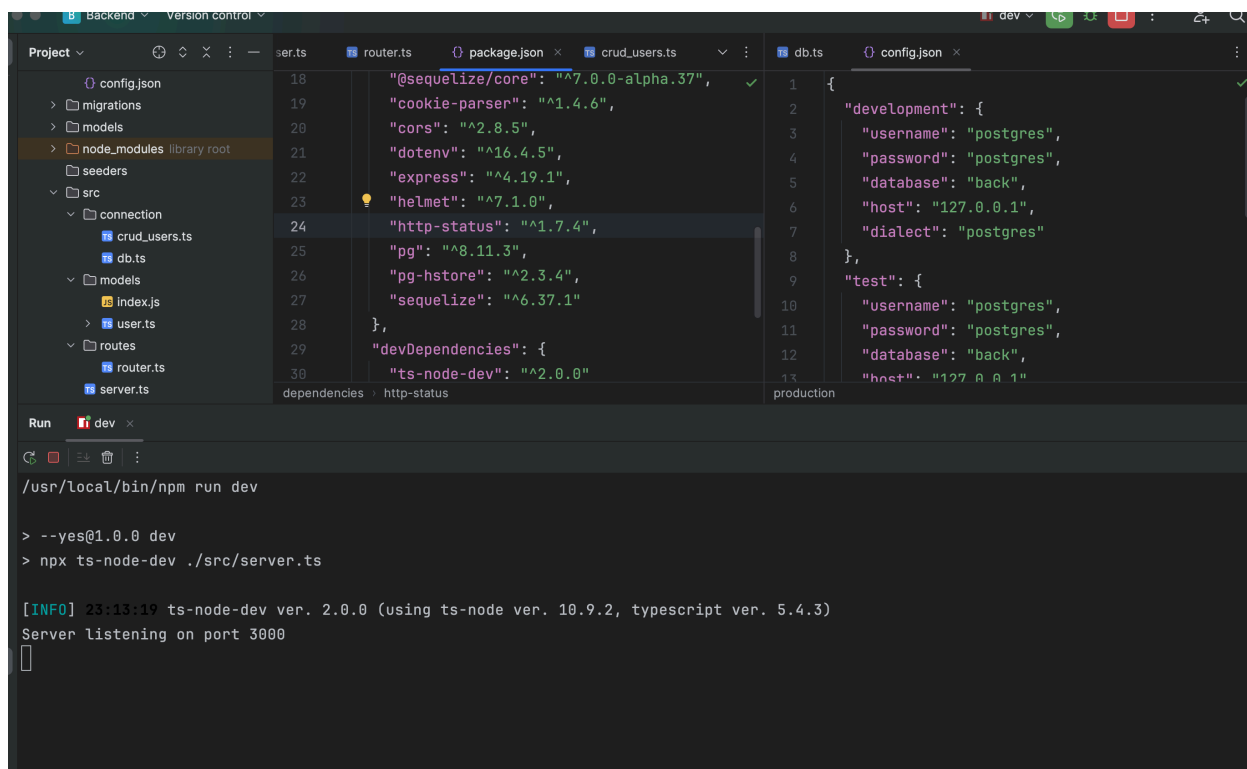
Рисунок 1 – Установка Sequelize и драйвера PostgreSQL

Далее была создана база данных back, описана модель пользователя и проведены миграции (Рисунок 2).

```
(base) ekaterinamalutina@Air-Ekaterina-4 Backend % npx sequelize-cli init  
Need to install the following packages:  
sequelize-cli@6.6.2  
Ok to proceed? (y) y  
  
Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.1]  
  
Created "config/config.json"  
Successfully created models folder at "/Users/ekaterinamalutina/WebstormProjects/Backend/models".  
Successfully created migrations folder at "/Users/ekaterinamalutina/WebstormProjects/Backend/migrations".  
Successfully created seeders folder at "/Users/ekaterinamalutina/WebstormProjects/Backend/seeders".  
(base) ekaterinamalutina@Air-Ekaterina-4 Backend % npx sequelize-cli model:generate --name User --attributes id:integer,name:string,email:string,tax_payer:boolean,residence:string  
  
Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.1]  
  
New model was created at /Users/ekaterinamalutina/WebstormProjects/Backend/models/user.js .  
New migration was created at /Users/ekaterinamalutina/WebstormProjects/Backend/migrations/20240321200346-create-user.js .  
(base) ekaterinamalutina@Air-Ekaterina-4 Backend % 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"
```

Рисунок 2 – Настройка базы данных

Далее был исправлен файл конфигурации и запущен сервер (Рисунок 3).



The screenshot shows a code editor with several files open. The 'package.json' file is visible, showing dependencies like 'sequelize', 'cookie-parser', 'cors', 'dotenv', 'express', 'helmet', 'http-status', 'pg', 'pg-hstore', and 'sequelize'. The 'config.json' file is also open, showing development and test configurations for a PostgreSQL database. The terminal window at the bottom shows the command 'npm run dev' being executed, and the server is listening on port 3000.

```
dependencies: {
  "sequelize/core": "^7.0.0-alpha.37",
  "cookie-parser": "^1.4.6",
  "cors": "^2.8.5",
  "dotenv": "^16.4.5",
  "express": "^4.19.1",
  "helmet": "^7.1.0",
  "http-status": "^1.7.4",
  "pg": "^8.11.3",
  "pg-hstore": "^2.3.4",
  "sequelize": "^6.37.1"
},
devDependencies: {
  "ts-node-dev": "^2.0.0"
}
```

```
{
  "development": {
    "username": "postgres",
    "password": "postgres",
    "database": "back",
    "host": "127.0.0.1",
    "dialect": "postgres"
  },
  "test": {
    "username": "postgres",
    "password": "postgres",
    "database": "back",
    "host": "127.0.0.1"
  }
}
```

```
/usr/local/bin/npm run dev

> --yes@1.0.0 dev
> npx ts-node-dev ./src/server.ts

[INFO] 23:13:19 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.4.3)
Server listening on port 3000
```

Рисунок 3 – Исправленные файлы конфигурации и зависимости

Далее были написаны функции работы с данными с базой данных и функциональность была протестирована в программе Postman (Рисунки 4-7 на страницах 3-5).

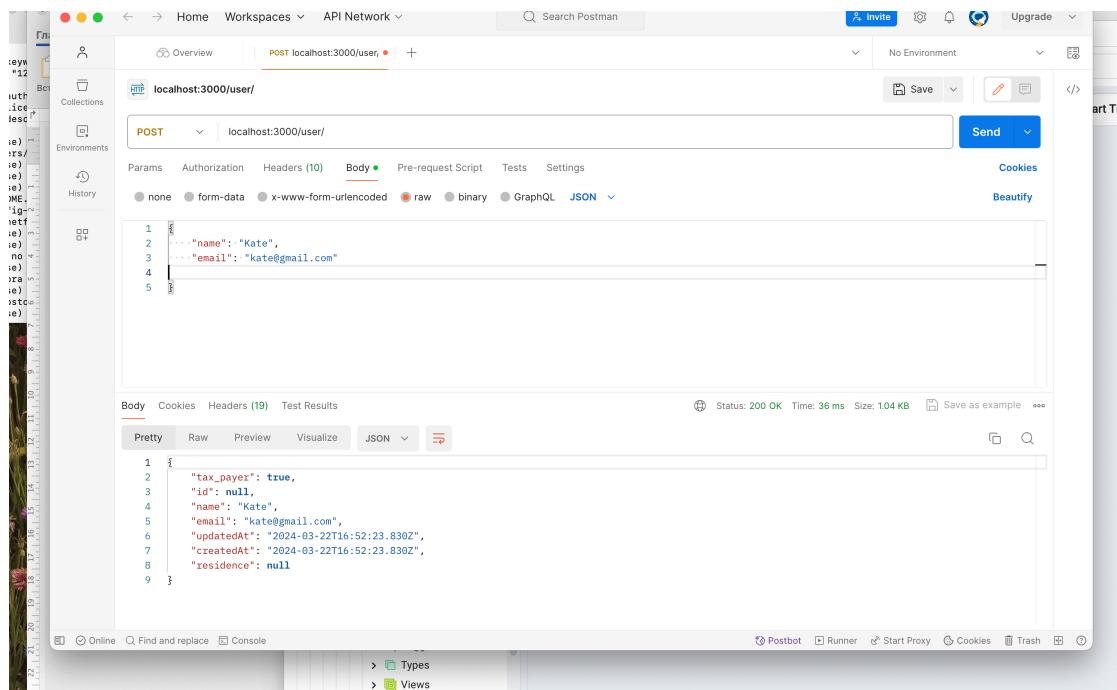


Рисунок 4 – Post запрос

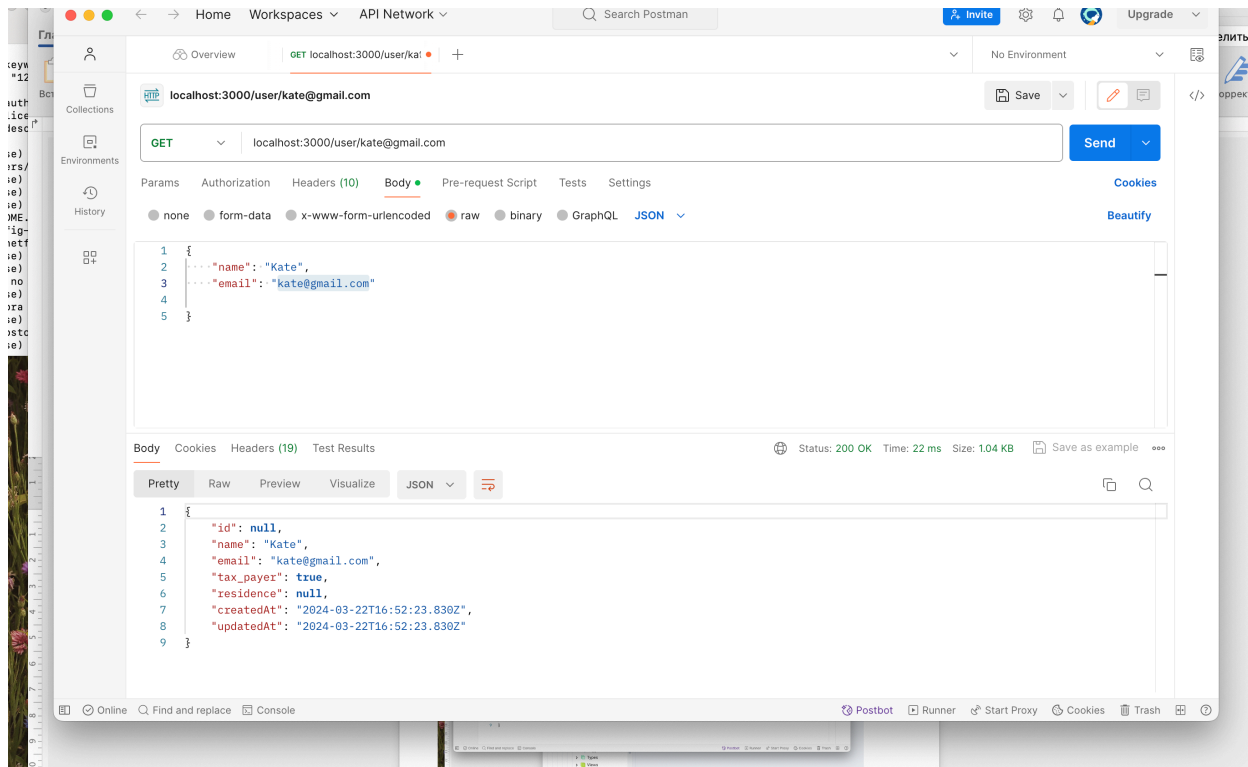


Рисунок 5 – Get запрос

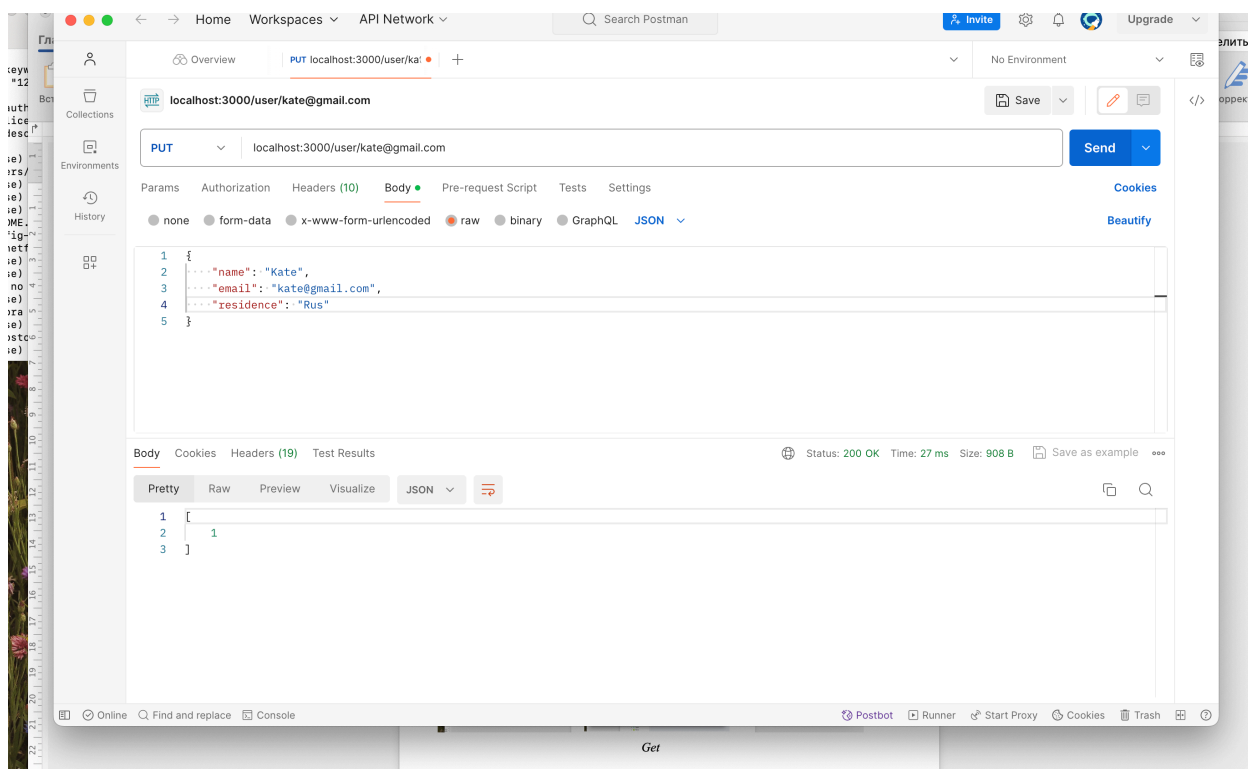


Рисунок 6 – Put запрос

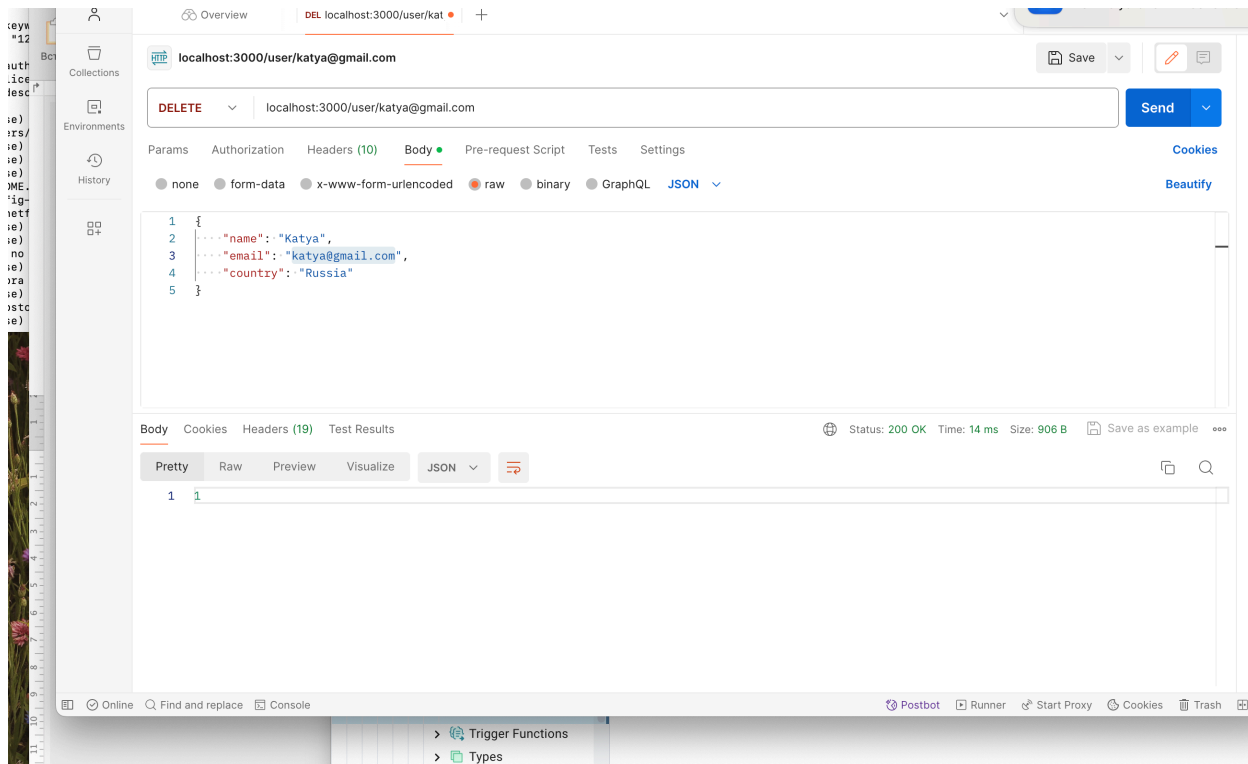


Рисунок 7 – Delete запрос

Выводы.

Научилась работать с express и sequelize, написала пользовательские обработчики запросов, а также были установлены дополнительные библиотеки и настроены.