

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)**

Факультет Инфокоммуникационных технологий (ИКТ)

Образовательная программа Мобильные и сетевые технологии

О Т Ч Е Т

по дисциплине: Бек-энд разработка
по Домашней работе 2

Тема задания: «Знакомство с ORM Sequelize»

Обучающийся: Кулагина Светлана Викторовна, группа К33412
Преподаватель: Добряков Д.И.

Оценка: _____

Дата: «__» _____ 2023 г.

Санкт-Петербург
2023

ЗАДАЧИ

- 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

Модель пользователя в файле `user.js`, созданная с помощью команды `npx sequelize-cli model:generate --name User --attributes username:string,firstName:string,lastName:string,email:string,password:string`

```
models > JS user.js > <unknown> > exports
1 'use strict';
2 const {
3   Model
4 } = require('sequelize');
5 module.exports = (sequelize, DataTypes) => {
6   class User extends Model {
7     /**
8      * Helper method for defining associations.
9      * This method is not a part of Sequelize lifecycle.
10     * The `models/index` file will call this method automatically.
11     */
12     static associate(models) {
13       // define association here
14     }
15   }
16   User.init({
17     username: DataTypes.STRING,
18     firstName: DataTypes.STRING,
19     lastName: DataTypes.STRING,
20     email: DataTypes.STRING,
21     password: DataTypes.STRING
22   }, {
23     sequelize,
24     modelName: 'User',
25   });
26   return User;
27 };
```

CRUD – методы для получения, изменения информации о пользователе и удаления пользователей

```

JS index.js > app.get('/user/:id') callback
1 const express = require('express')
2 const app = express()
3 const port = 3000
4 const db = require('./models')
5
6 app.use(express.json());
7
8 app.get('/', (req, res) => {
9   return res.send('Hello World!')
10 })
11
12 app.get('/users', async (req, res) => {
13   const users = await db.User.findAll()
14   return res.send(users)
15 })
16
17 app.get('/user/:id', async (req, res) => {
18   const user = await db.User.findById(req.params.id)
19   return res.send(user)
20 })
21
22 app.post('/user', async (req, res) => {
23   const user = await db.User.create(req.body)
24   return res.send(user.toJSON())
25 })
26

```

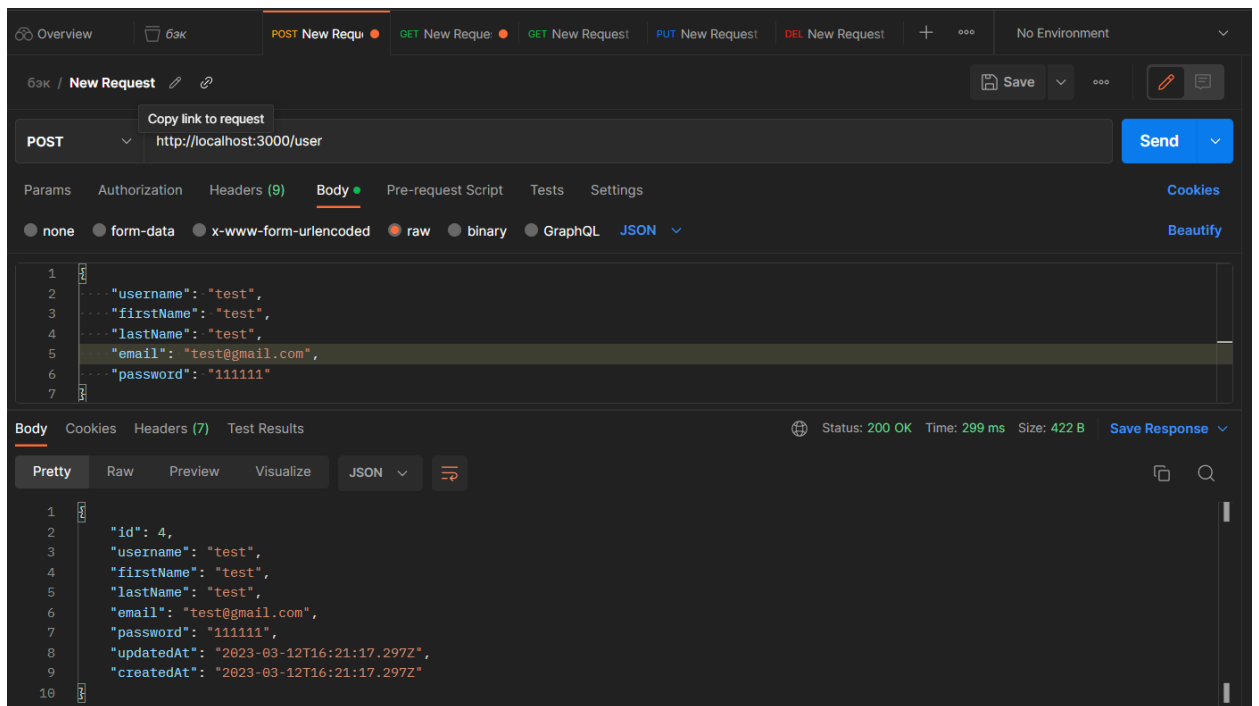
```

27 app.put('/user/:id', async (req, res) => {
28   const user = await db.User.update(req.body, {where: {id: req.params.id}})
29   const iduser = req.params.id
30   return res.send(`Информация о пользователе ${iduser} была обновлена`)
31 })
32
33 app.delete('/user/:id', async (req, res) => {
34   const user = await db.User.destroy({where: {id: req.params.id}})
35   const iduser = req.params.id
36   return res.send(`Пользователь под номером ${iduser} удален из базы данных`)
37 })
38
39 app.listen(port, () => {
40   console.log(`Example app listening on port ${port}`)
41 })

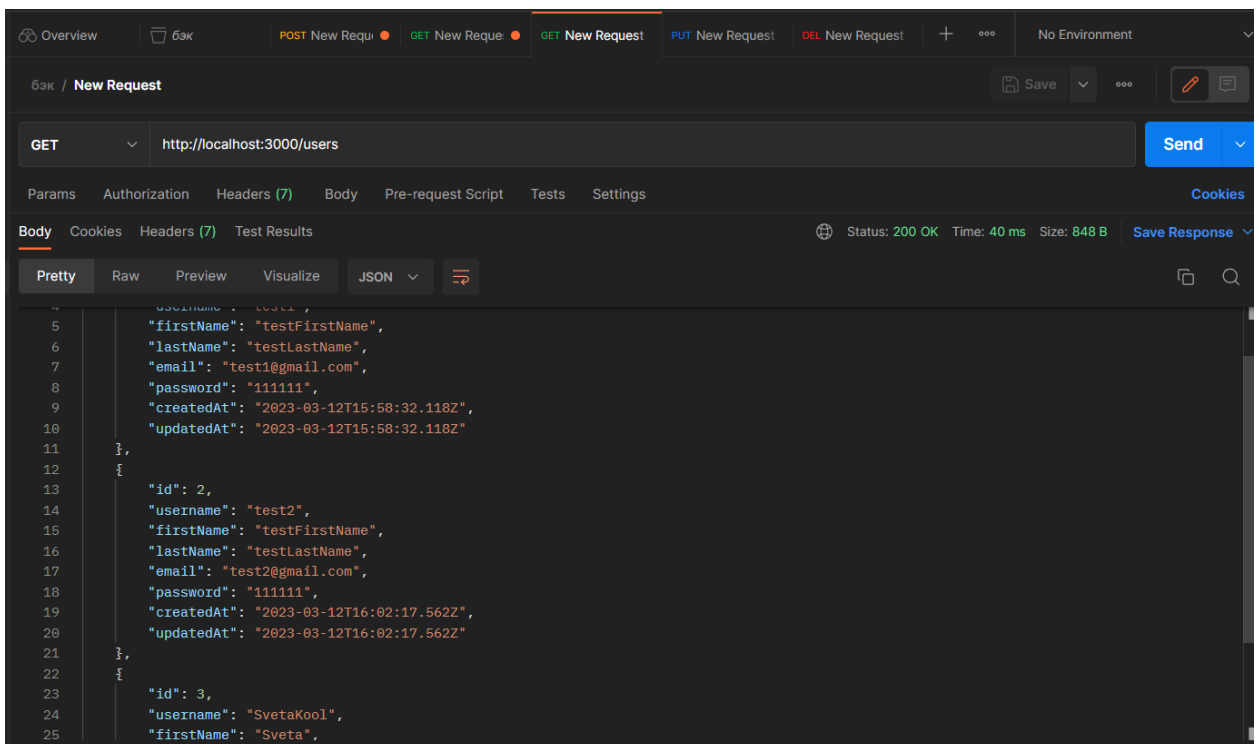
```

Тестирование CRUD -методов с помощью POSTMAN

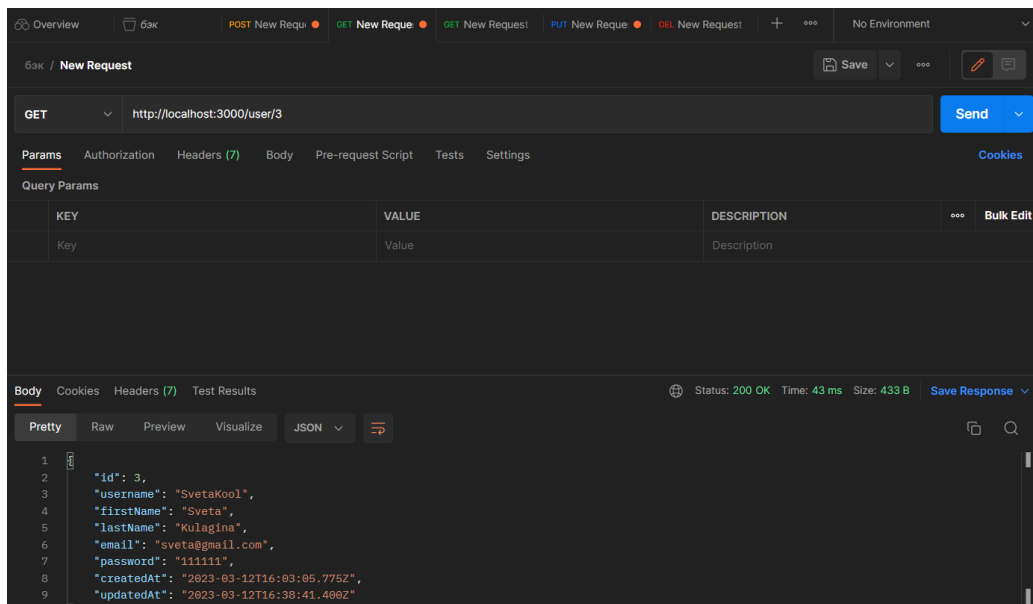
POST запрос на создание пользователя



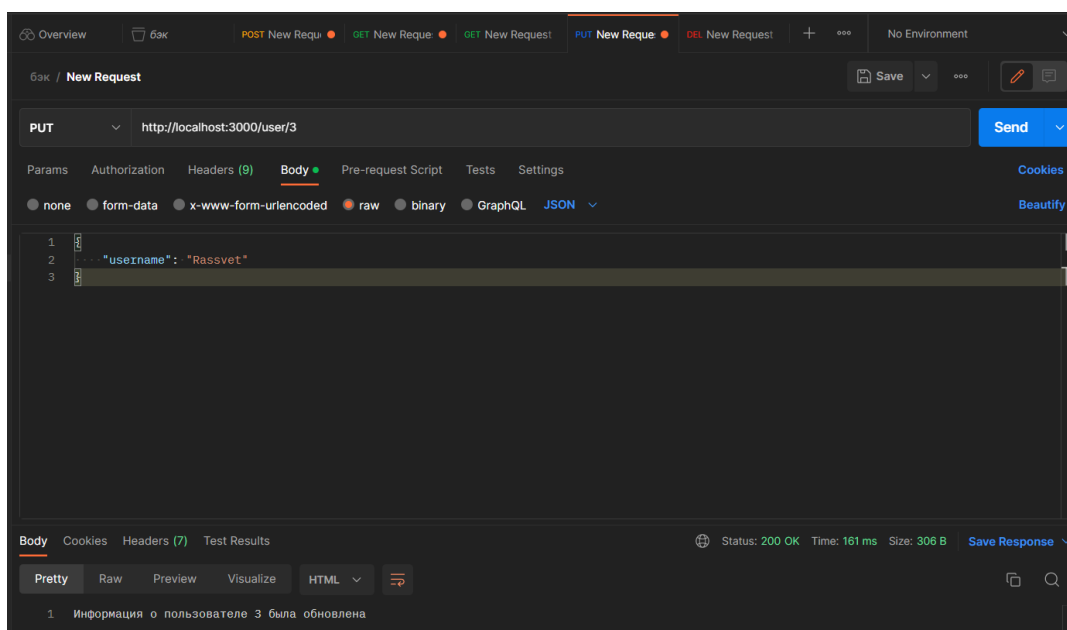
GET запрос на получение всех пользователей



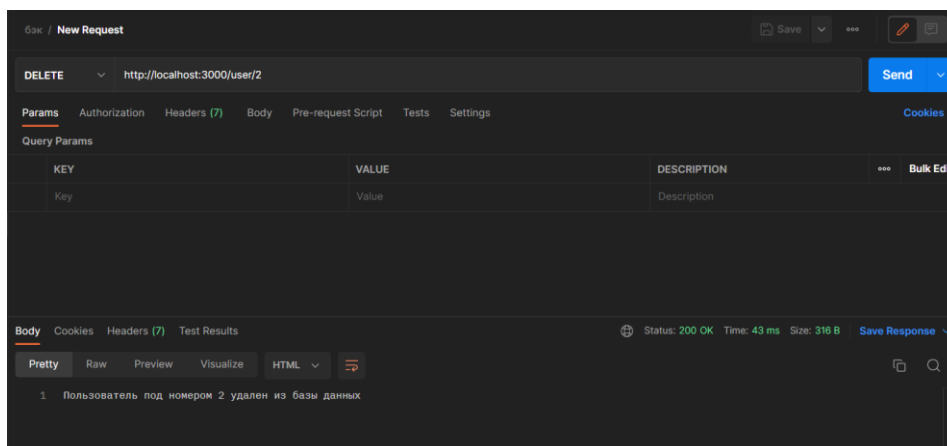
GET запрос на получение информации о конкретном пользователе по id



PUT запрос на изменение информации о конкретном пользователе по id



DELETE запрос на удаление конкретного пользователя по id



ВЫВОД

По результатам данной домашней работы я научилась создавать модели с помощью sequelize, а также научилась создавать и тестировать CRUD – методы с помощью express. Sequelize и POSTMAN