**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-енд разработка
по Домашней работе 2

Тема задания: «Знакомство с ORM Sequelize»

Обучающийся: Кириллова В.Е., К33402
Преподаватель: Добряков Д.И.

Оценка: ____

Дата: «__»_____2023 г.

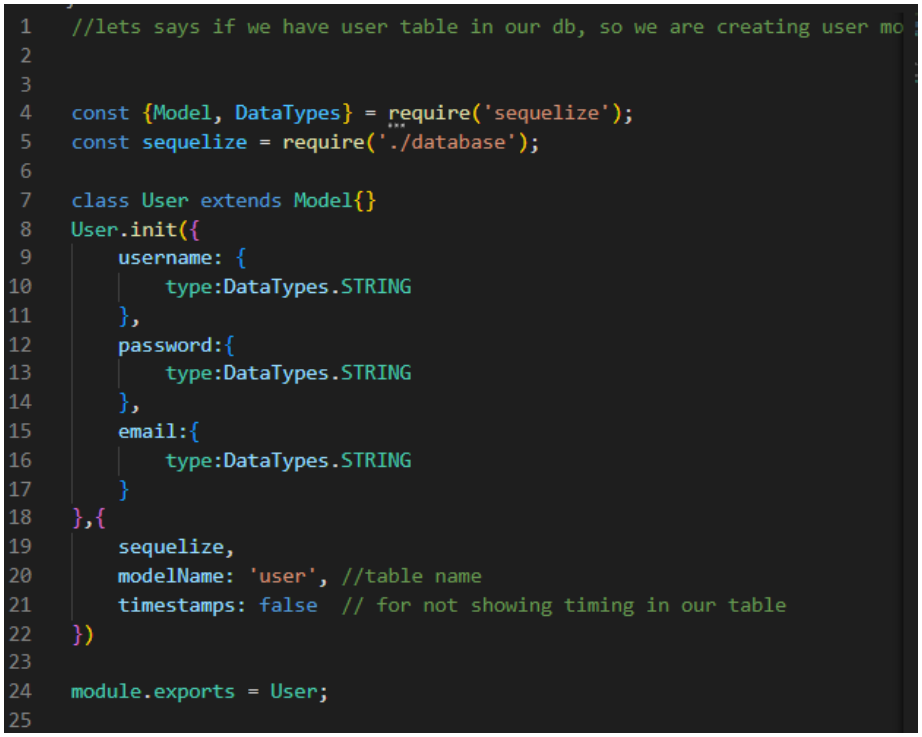
Санкт-Петербург
2023

## 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



```
//lets says if we have user table in our db, so we are creating user mo


const {Model, DataTypes} = require('sequelize');
const sequelize = require('./database');

class User extends Model{}
User.init({
    username: {
        type:DataTypes.STRING
    },
    password:{
        type:DataTypes.STRING
    },
    email:{
        type:DataTypes.STRING
    }
},{
    sequelize,
    modelName: 'user', //table name
    timestamps: false  // for not showing timing in our table
})

module.exports = User;

```

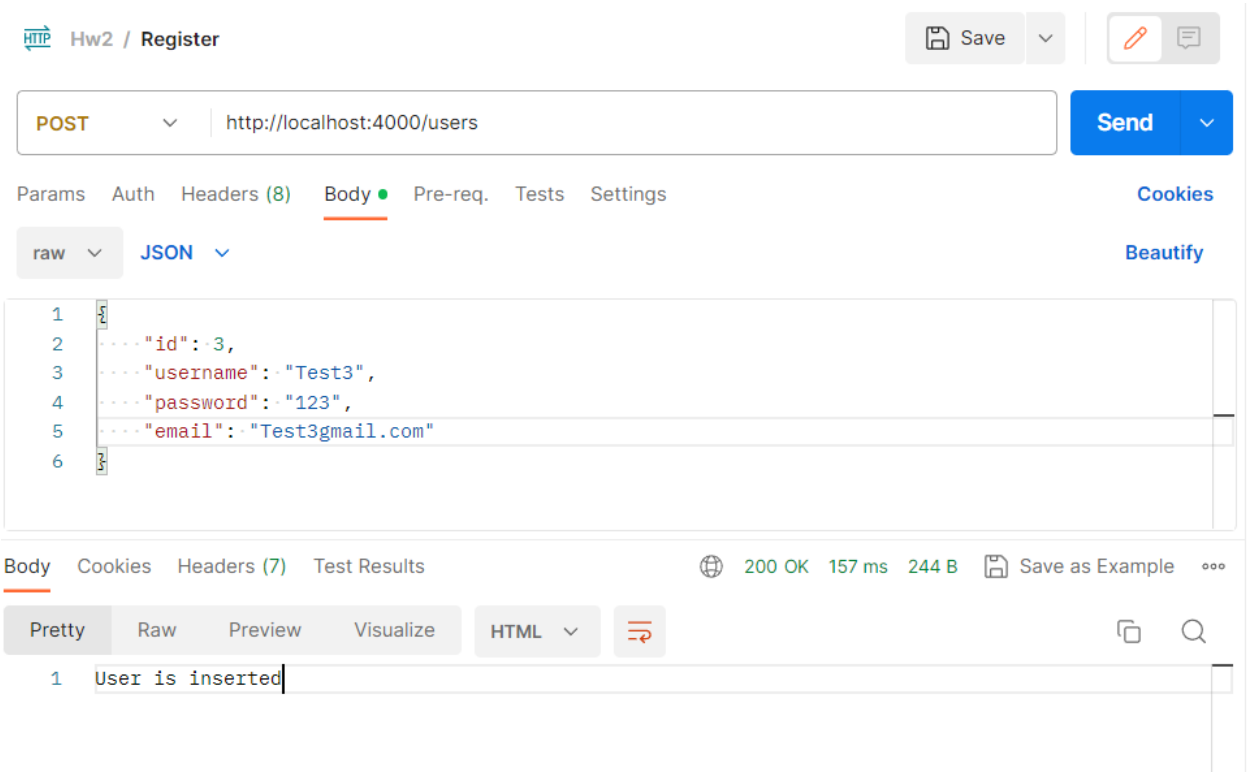
CRUD – методы для получения, изменения информации о пользователе и удаления пользователей

index.js > ...

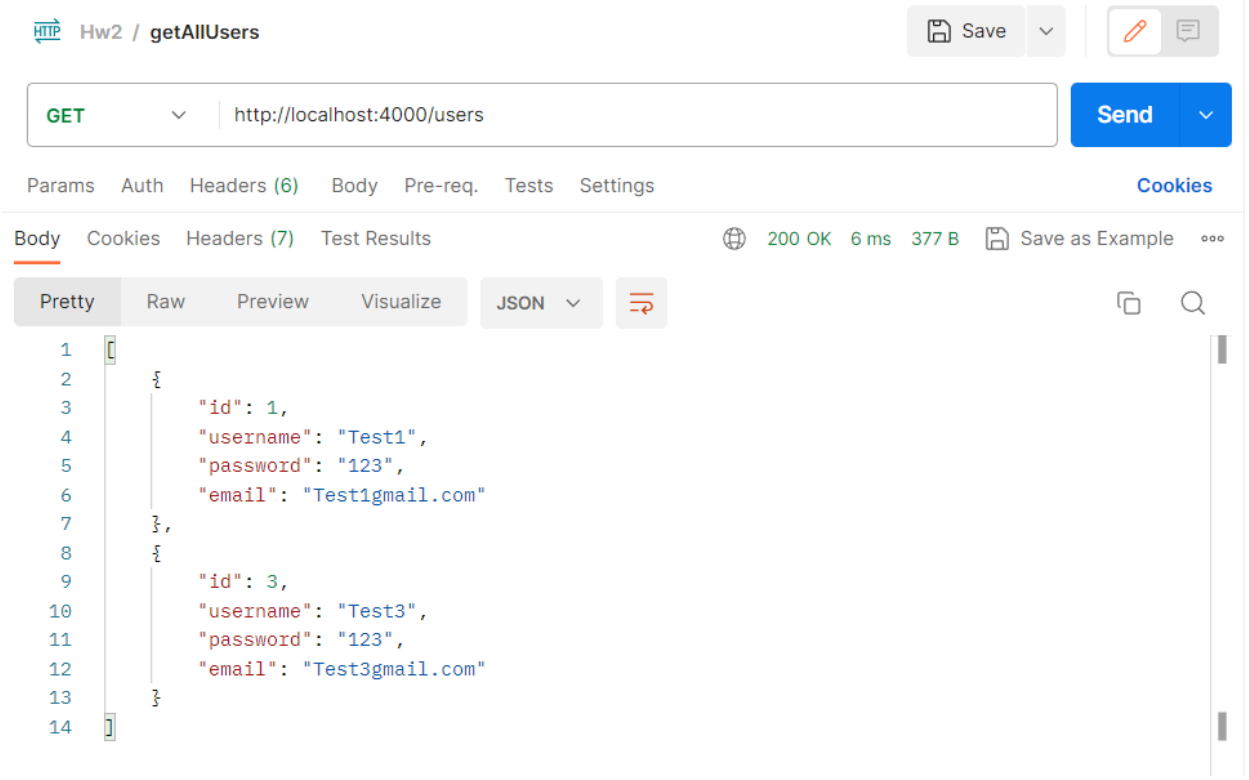
```
const express = require("express");
const port = 4000;

const sequelize = require('./database');
const User = require('./User');
sequelize.sync({force: true}).then(()=> console.log("db is ready"))
//force: true. is used here to delete old data from table and create table again e
const app = express()
app.use(express.json()); // this to make express aware what data is coming. unless
app.post('/users', async (req,res)=>{
    await User.create(req.body)
        res.send("User is inserted")
});
//Show All Users
app.get('/users', async (req, res)=>{
    const myuser = await User.findAll()
        res.send(myuser);
    })
//Find By Id
app.get('/users/:id',async (req,res)=>{
    const requestedId = req.params.id;
    const requestedUser = await User.findOne({where: {id: requestedId} })
    res.send(requestedUser)
})

//Update By Id
app.put('/users/:id',async (req,res)=>{
    const requestedId = req.params.id;
    const requestedUser = await User.findOne({where: {id: requestedId} })
    requestedUser.username = req.body.username;
    await requestedUser.save()
    res.send("Updated")
})

//Delete By Id
app.delete('/users/:id',async (req,res)=>{
    const requestedId = req.params.id;
    const requestedUser = await User.destroy({where: {id: requestedId} })
    res.send("deleted")
})

app.listen(port, ()=>{
    console.log(`Server is running on Port ${port}`)
})
```

Тестирование CRUD -методов с помощью POSTMAN

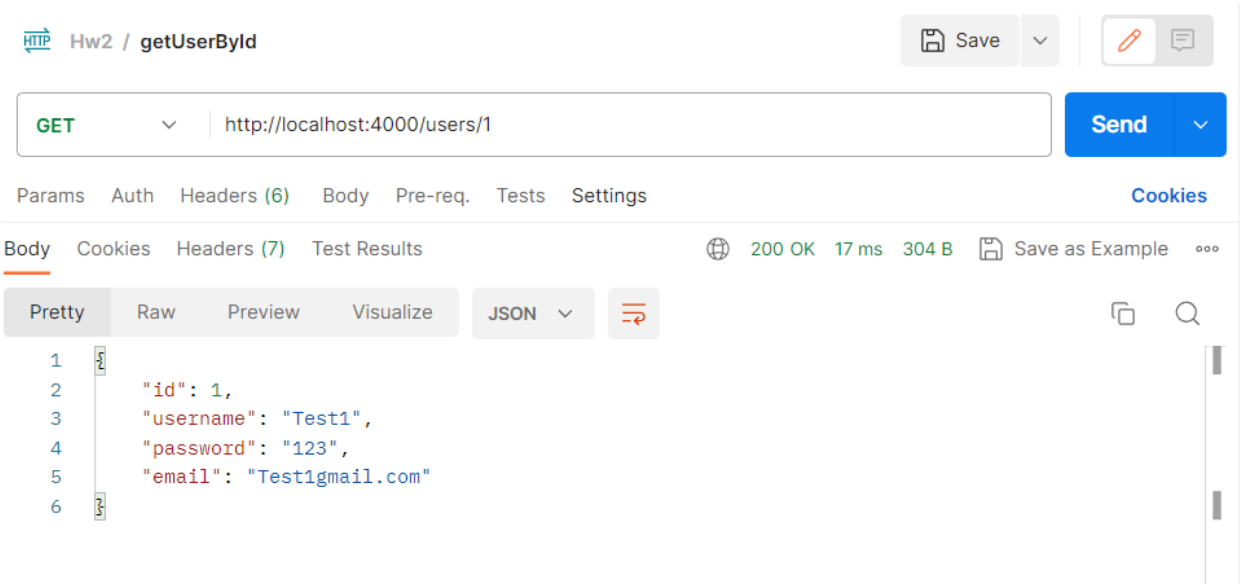
**POST** запрос на создание пользователя



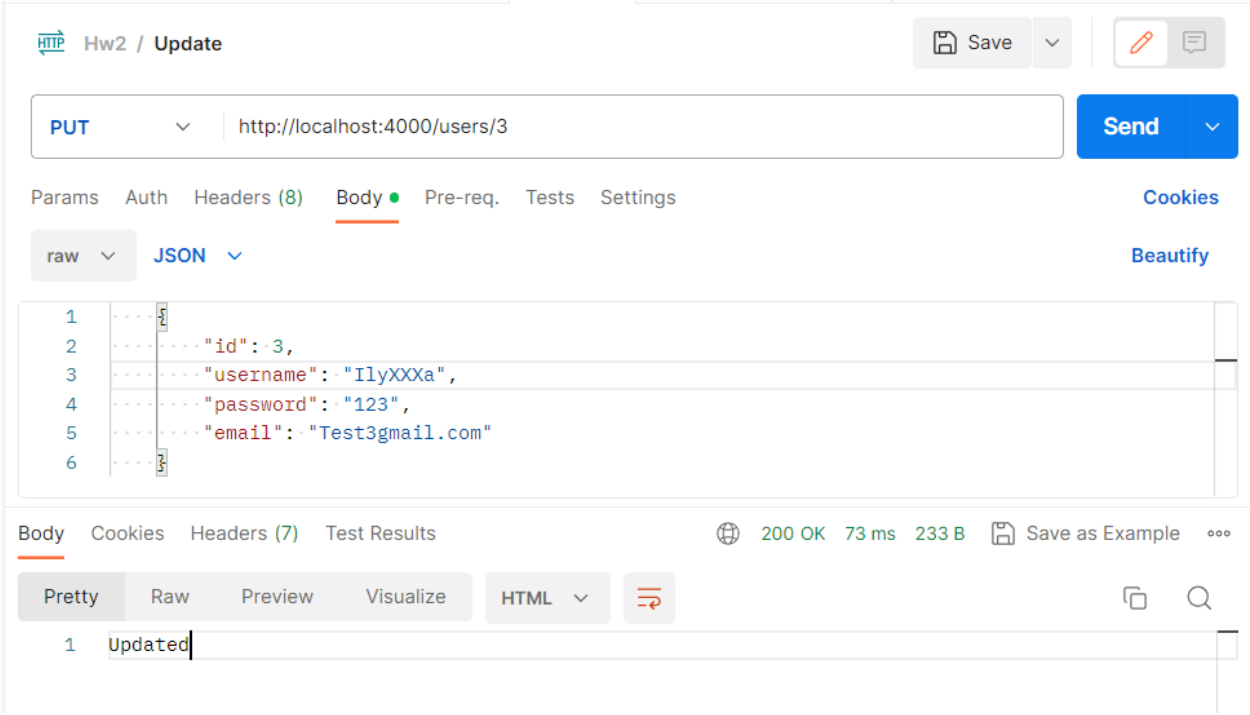

**GET** запрос на получение всех пользователей



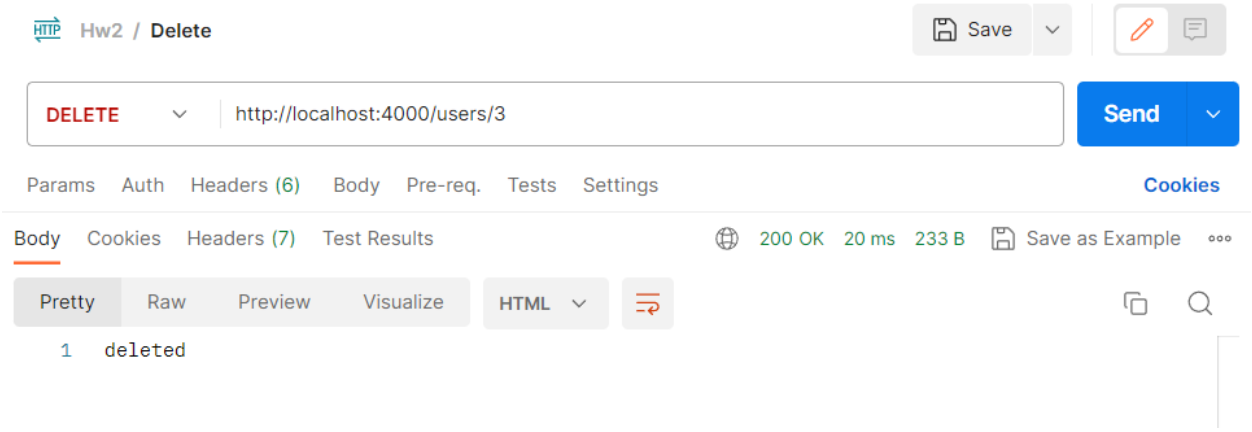

**GET** запрос на получение информации о конкретном пользователе по id




**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