Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по домашней работе №2

по дисциплине **«Бэкенд-разработка»**

Выполнил: Холодов-Воронцов А. А.

Факультет: Инфокоммуникационных технологий

Группа: K33412

Проверил: Добряков Д. И.

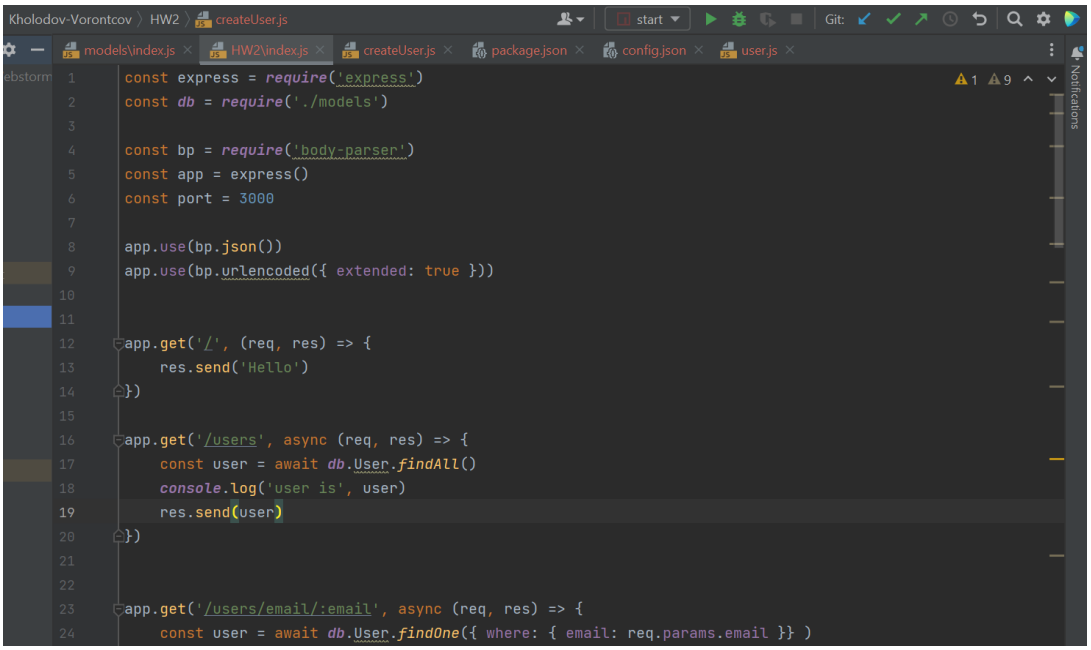
Санкт-Петербург 2023

Задание

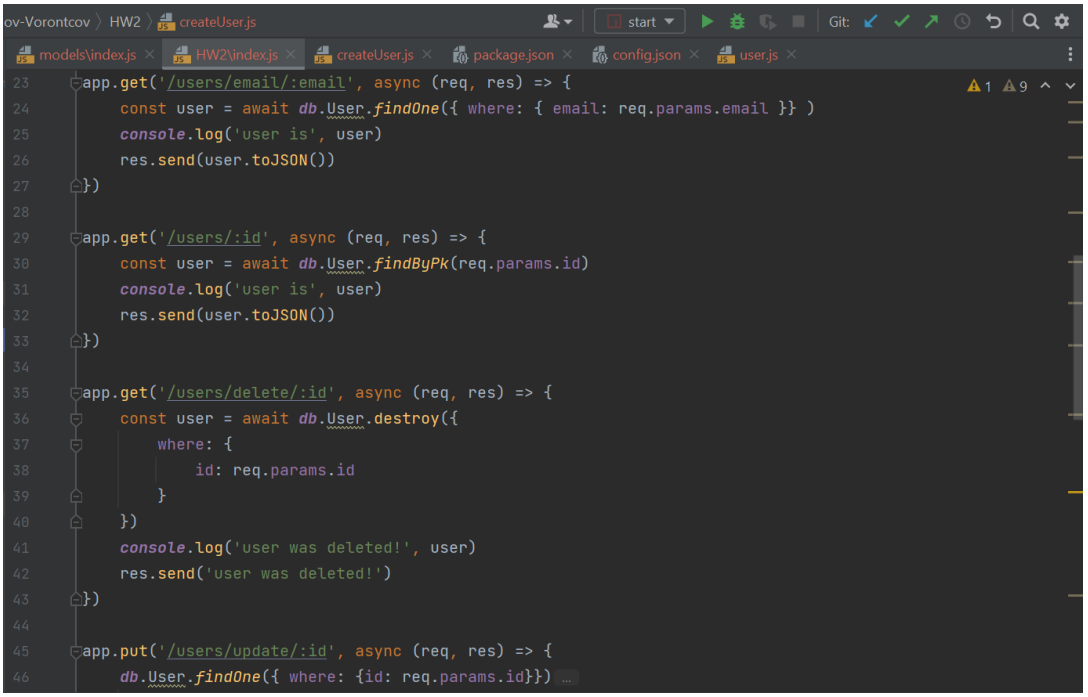
- Продумать свою собственную модель пользователя
- Реализовать набор из CRUD-методов для работы с пользователями средствами Express + Sequelize
- Написать запрос для получения пользователя по id/email

Ход работы

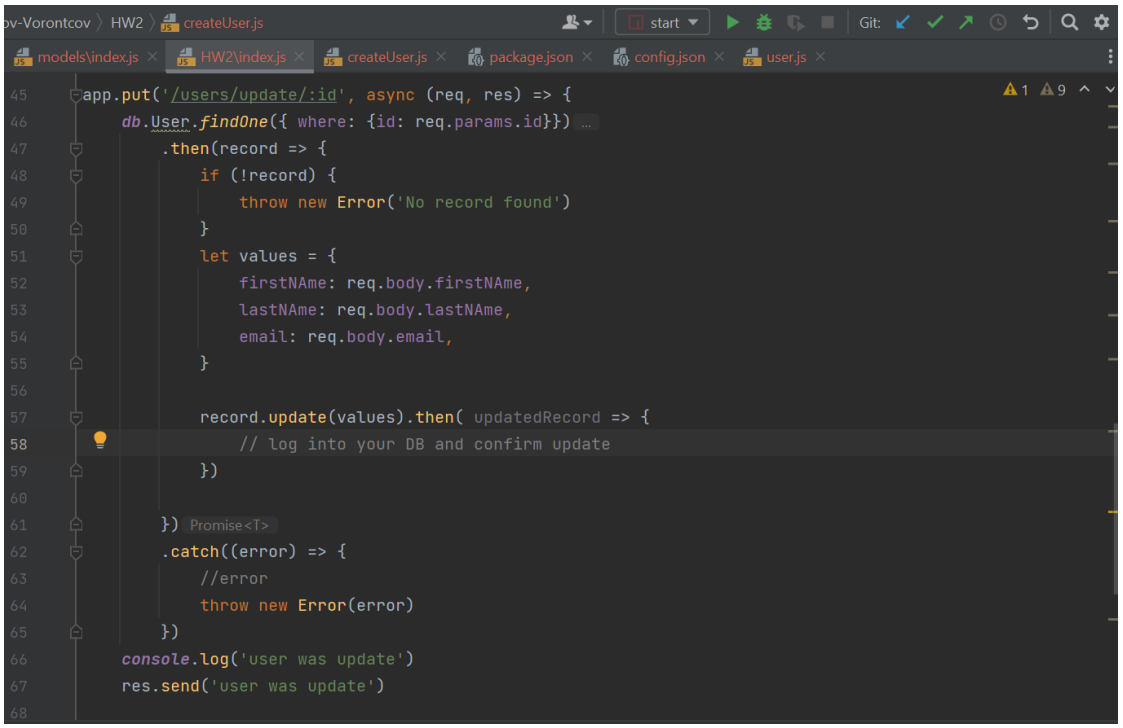
1. index.js



```
const express = require('express')
const db = require('./models')

const bp = require('body-parser')
const app = express()
const port = 3000

app.use(bp.json())
app.use(bp.urlencoded({ extended: true }))


app.get('/', (req, res) => {
    res.send('Hello')
})

app.get('/users', async (req, res) => {
    const user = await db.User.findAll()
    console.log('user is', user)
    res.send(user)
})


app.get('/users/email/:email', async (req, res) => {
    const user = await db.User.findOne({ where: { email: req.params.email }} )
```

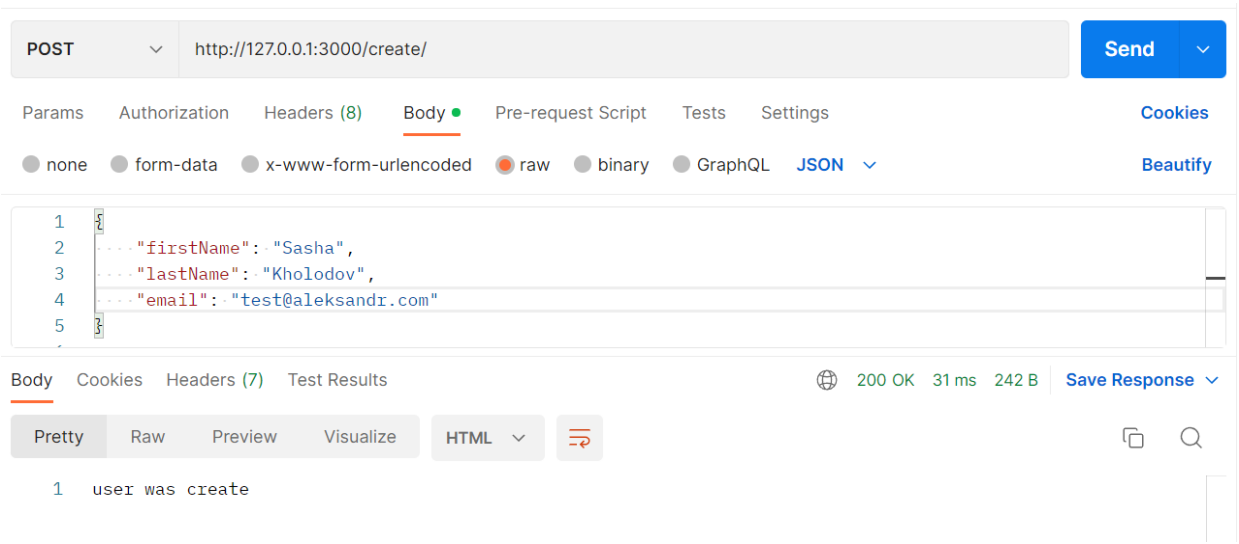


```
app.get('/users/email/:email', async (req, res) => {
    const user = await db.User.findOne({ where: { email: req.params.email }} )
    console.log('user is', user)
    res.send(user.toJSON())
})

app.get('/users/:id', async (req, res) => {
    const user = await db.User.findByPk(req.params.id)
    console.log('user is', user)
    res.send(user.toJSON())
})

app.get('/users/delete/:id', async (req, res) => {
    const user = await db.User.destroy({
        where: {
            id: req.params.id
        }
    })
    console.log('user was deleted!', user)
    res.send('user was deleted!')
})

app.put('/users/update/:id', async (req, res) => {
    db.User.findOne({ where: {id: req.params.id}})
```

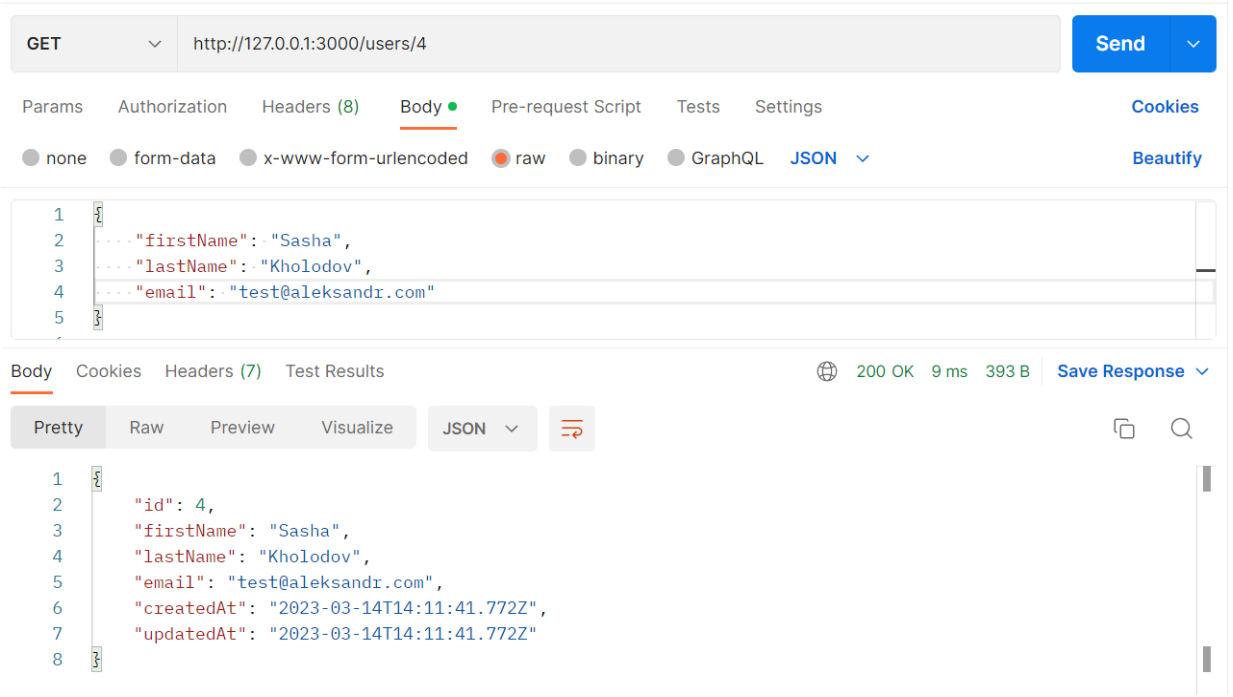


2. createUser.js

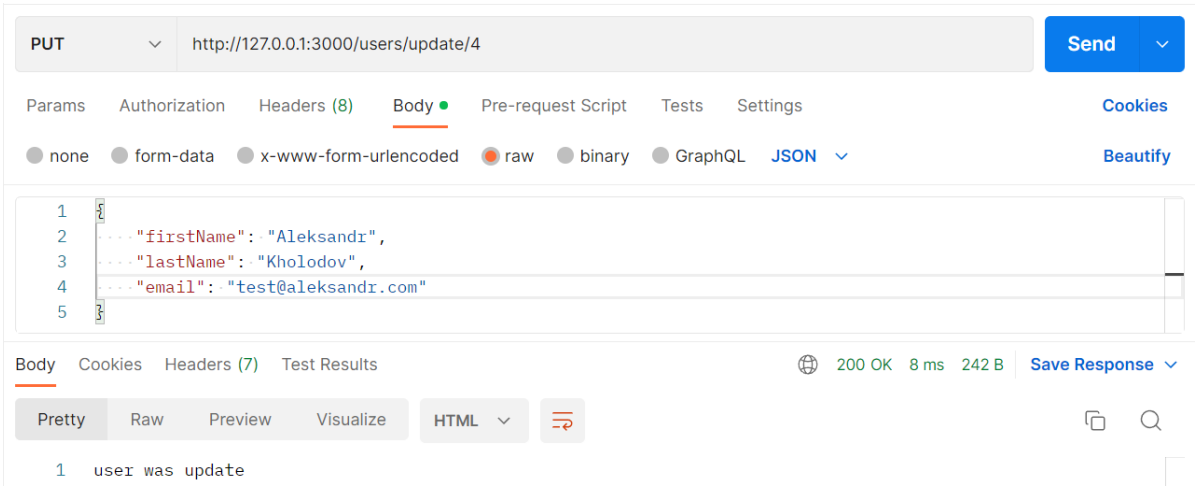
3. Создание пользователя:



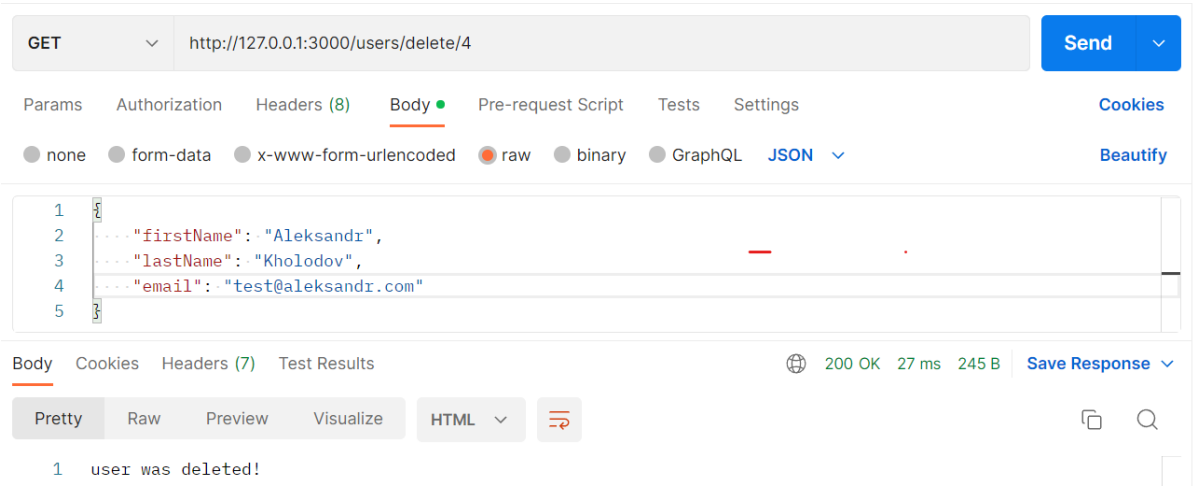
4. Получение пользователя по id:



5. Обновление данных пользователя по id:



6. Удаление пользователя по id:



Вывод

В ходе выполнения домашнего задания я научился создавать, получать, обновлять и удалять пользователей с помощью Node.js, используя express и sequelize.