САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа

Выполнил:

Олейникова Полина

Группа K33402

Проверил:
Добряков Д. И.

Санкт-Петербург

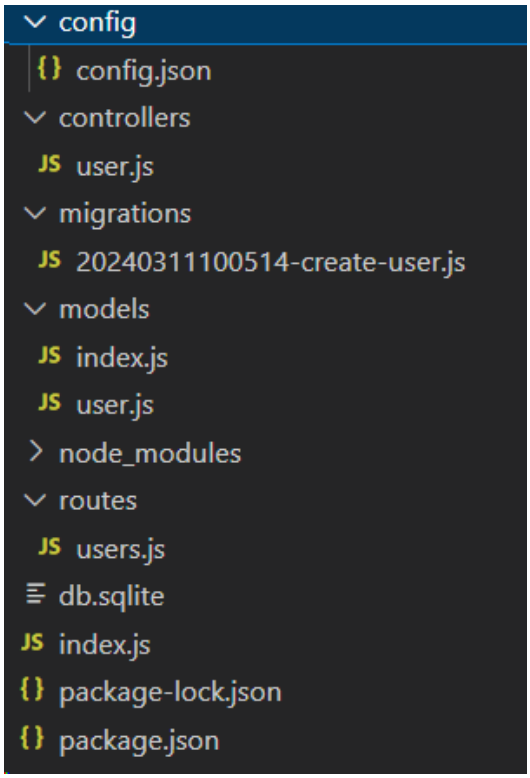
2024 г.

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

Структура папок:



Запуск сервера и определение маршрута для пользователей

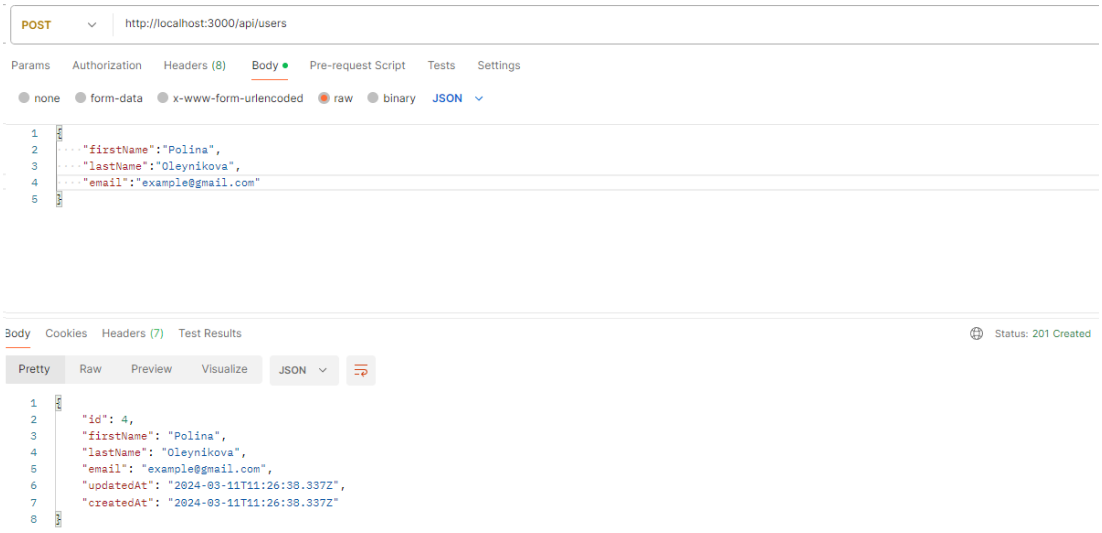
```
const express = require('express')
const users = require('./routes/users');

const app = express()
app.use(express.json());

const port = 3000

app.use('/api/users', users);

app.listen(port, () => {
    console.log(`Example app listening on port ${port}`)
})
```

Определение маршрутов к методам с пользователями

```
JS users.js
1   const express = require('express');
2   const router = express.Router();
3   const User = require('../controllers/user');
4
5   router.post('/', User.createUser);
6   router.get('/', User.getUsers);
7   router.get('/:id', User.getUser);
8   router.patch('/:id', User.updateUser);
9   router.delete('/:id', User.deleteUser);
10
11  module.exports = router;
```

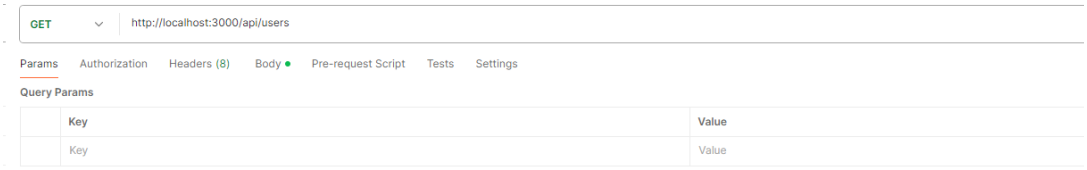
Создания пользователя

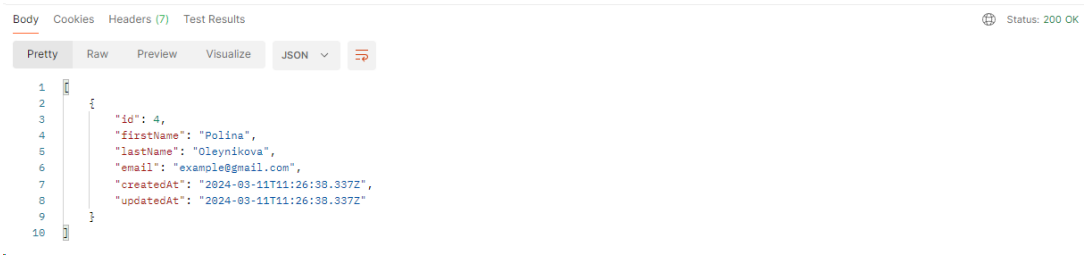
```
exports.createUser = async (req, res) => {
    try {
        const user = await db.User.create(req.body);
        return res.status(201).send(user);
    } catch (error) {
        return res.status(500).json({ message: error.message });
    }
}
```

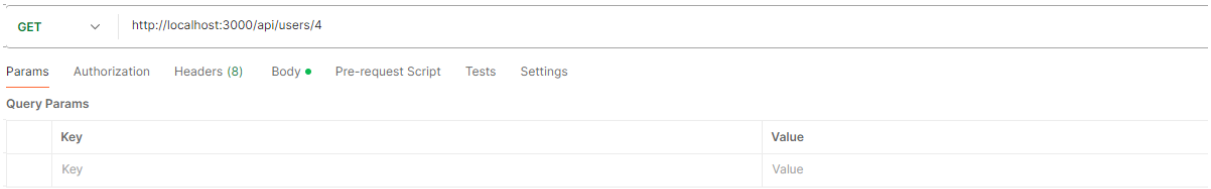


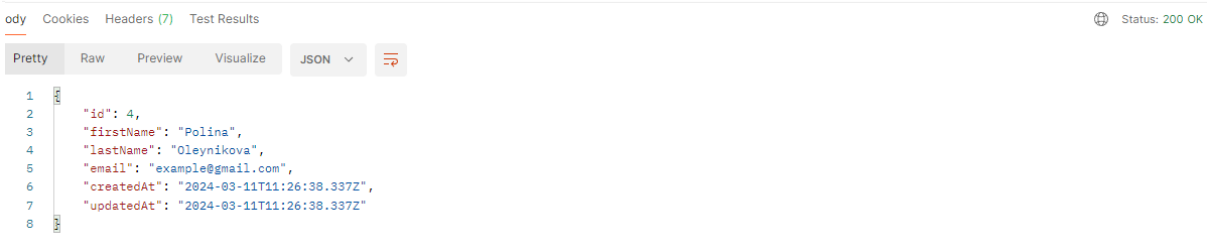
Получение пользователей\пользователя

```
exports.getUsers = async (req, res) => {
    const users = await db.User.findAll()
    return res.send(users)
}

exports.getUser = async (req, res) => {
    const user = await db.User.findByPk(req.params.id)
    if (user) {
        return res.send(user)
    }
    return res.status(404).send({ message: "user is not found" })
}
```



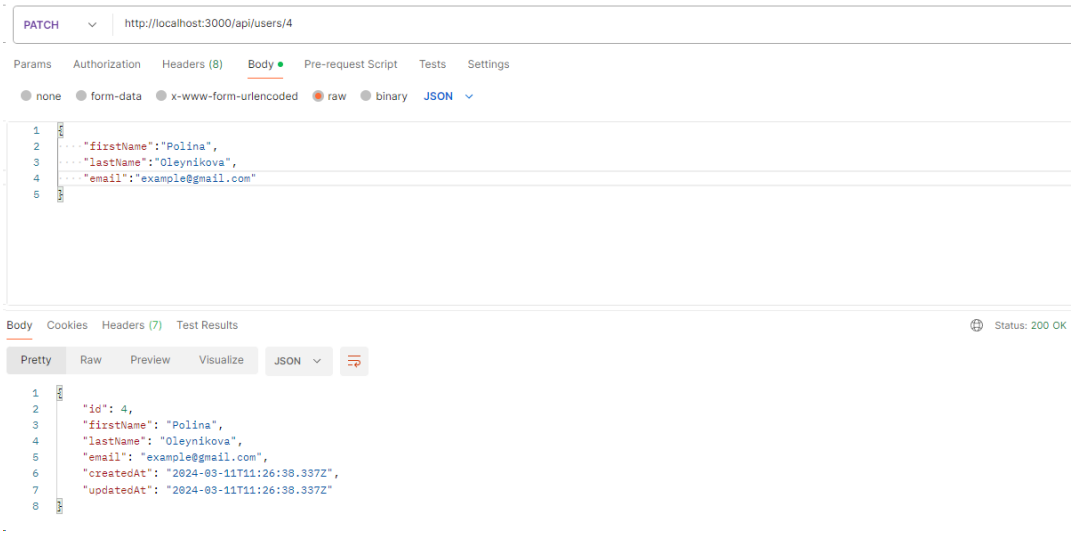






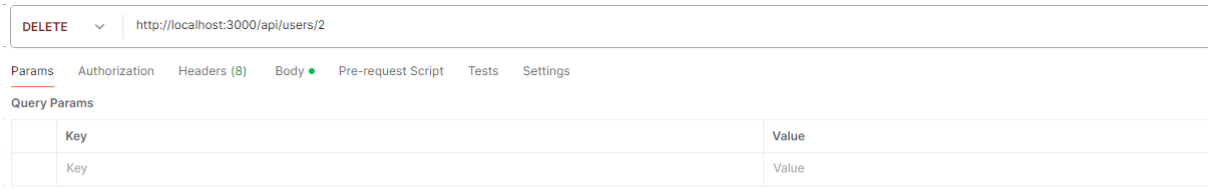
Изменение пользователя

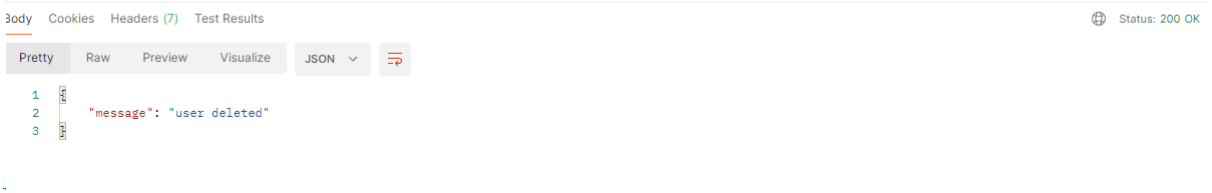
```
exports.updateUser = async (req, res) => {
    const user = await db.User.findByPk(req.params.id);
    if (!user) {
        return res.status(404).send({ message: "user is not found" });
    }

    try {
        const updatedUser = await user.update(req.body);
        return res.send(updatedUser);
    } catch (e) {
        return res.status(500).json({ message: error.message });
    }
}
```



Удаление пользователя

```
exports.deleteUser = async (req, res) => {
    const user = await db.User.destroy({
        where: {
            id: req.params.id
        }
    });
    if (user) {
        return res.send({ message: "user deleted" })
    }
    return res.status(404).send({ message: "user is not found" })
}
```





Вывод

В ходе данной работы был реализован набор из CRUD-методов для работы с пользователями средствами Express + Sequelize.