САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа 2

Выполнил:

Селезнев Илья

Группа К33412

Проверил: Добряков Д. И.

Санкт-Петербург

2023 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

Модель пользователя:

```
}, options {
    defaultScope: {
        // Do not return salt and password
        attributes: {exclude: ['password', 'salt']},
    },
    hooks: {
        beforeCreate: (instance M, options ::.) => {
            instance.salt = crypto.randomBytes( size 16).toString( encoding: 'base64');
        }
     }, sequelize,
    modelName: 'User',
   });
   return User;
};
```

Представления:

```
const express = require('express');
const bodyParser = require('body-parser')
const db = require('./models');

const app = express();
const port = 5000;

// Parse application/x-www-form-urlencoded
ap.use(bodyParser.urlencoded({ extended: false }))

// Parse application/json
app.use(bodyParser.json())

app.use(bodyParser.json())

app.use(bodyParser.json())

const user = await db.User.findByPk(req.params.id);

if (user) {
    res.send(user.toJSON());
}

res.status(code 404).send(body: {"detail": "User not found"});
```

```
app.get('/api/users', async (req :..., res :Response<ResBody,LocalsObj> ) => {
    const users = await db.User.findAll();
    res.send(users);
}

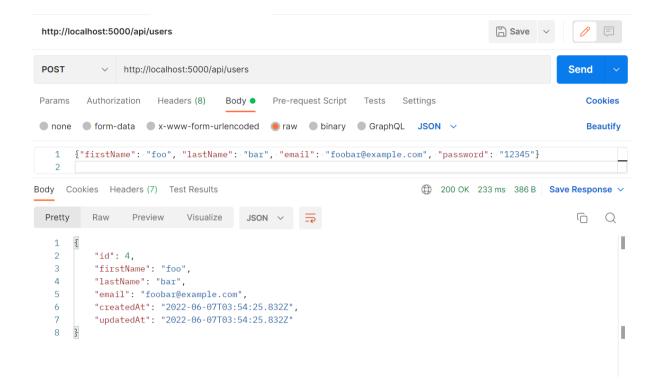
app.post( path: '/api/users', handlers: async (req :..., res :Response<ResBody,LocalsObj> ) => {
    try {
        const user = await db.User.create(req.body);
        await user.reload();
        res.send(user.toJSON());
    } catch (e) {
        console.log(e);
        res.status( code: 400).send( body: {"detail": "Failed to create user"});
    }
}
```

```
app.put( path '/ani/usens/id', handers async (req _ , res Response/ResBody LocalsCoby) ) => {
    const user = await db.User.findByPk(req.params.id);
    if (user) {
        user.update(req.body, {where: {id: req.params.id}});
        user.veload();
        res.send(user.toJSON());
    } catch (% {
        console.log(e);
        res.status(code 480).send(body {"detail": "Failed to update user"});
    }
} } else {
    res.status(code 484).send(body {"detail": "User not found"});
}

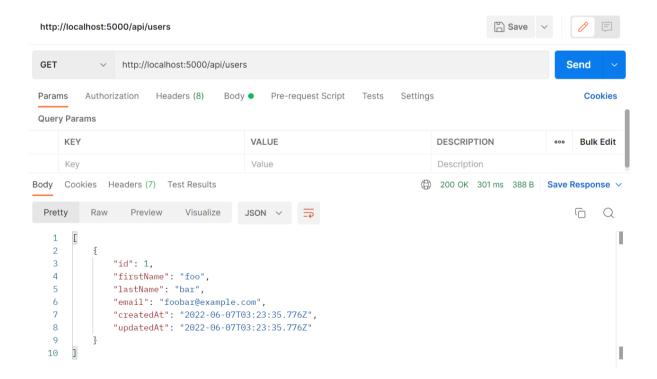
app.deLete( path: '/ani/usens/iid', handlers async (req: _, res Response/ResBody LocalsCoby) ) => {
    const user = await db.User.destroy({where: {id: req.params.id}})
    if (user) {
        res.send(body {"detail": "User deleted"});
    }
    res.status(code 404).send(body {"detail": "User not found"});
}

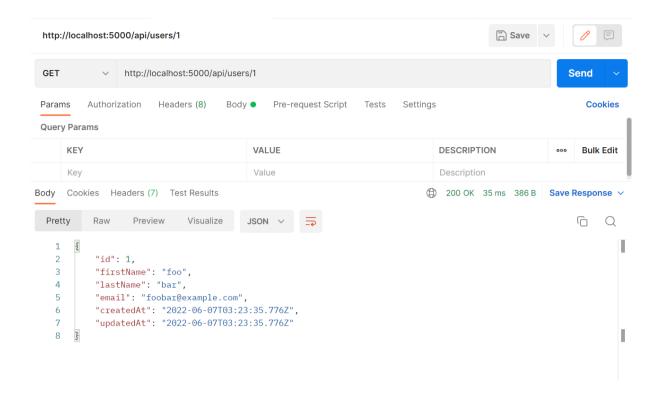
app.listen(port, hostname.() => {
    console.log(`Started app on port ${port}`');
}
```

Создание:



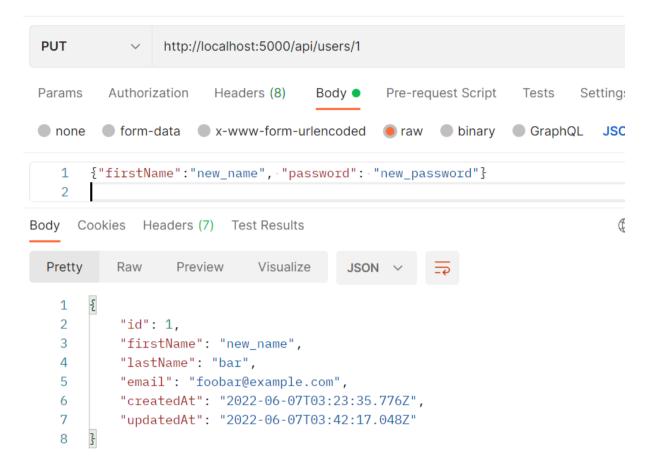
Чтение:



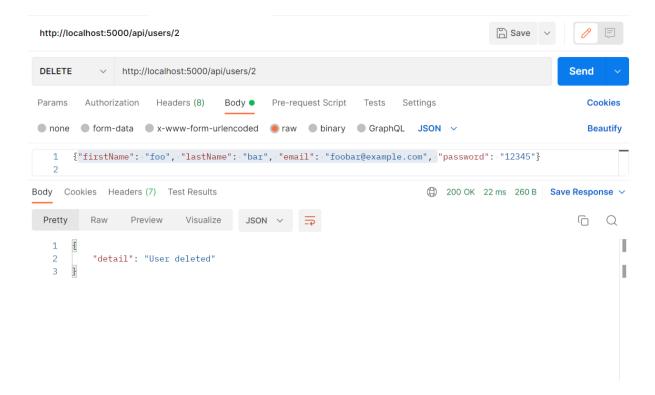


Изменение:

http://localhost:5000/api/users/1



Удаление:



Вывод

Я поработал с моделями средствами ORM Sequelize и созданию API средствами Express.