САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Backend-Разработка

Отчет

Практическая работа 2

Выполнил:
Бункута Натан

Группа:
К33412

Проверил:
Добрияков Д.И

Санкт-Петербург

2023 г.

Задание:

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

Выполнение работы

Models

models > JS index.js > ...

```
'use strict';

const fs = require('fs');
const path = require('path');
const Sequelize = require('sequelize');
const basename = path.basename(__filename);
const env = process.env.NODE_ENV || 'development';
const config = require(__dirname + '/../config/config.json')[env];
const db = {};

let sequelize;
if (config.use_env_variable) {
  sequelize = new Sequelize(process.env[config.use_env_variable], config);
} else {
  sequelize = new Sequelize(config.database, config.username, config.password, config);
}

fs
  .readdirSync(__dirname)
  .filter(file => {
    return (file.indexOf('.') !== 0) && (file !== basename) && (file.slice(-3) === '.js');
  })
  .forEach(file => {
    const model = require(path.join(__dirname, file))(sequelize, Sequelize.DataTypes);
    db[model.name] = model;
  });

Object.keys(db).forEach(modelName => {
  if (db[modelName].associate) {
    db[modelName].associate(db);
  }
});
```

models > JS user.js > ...

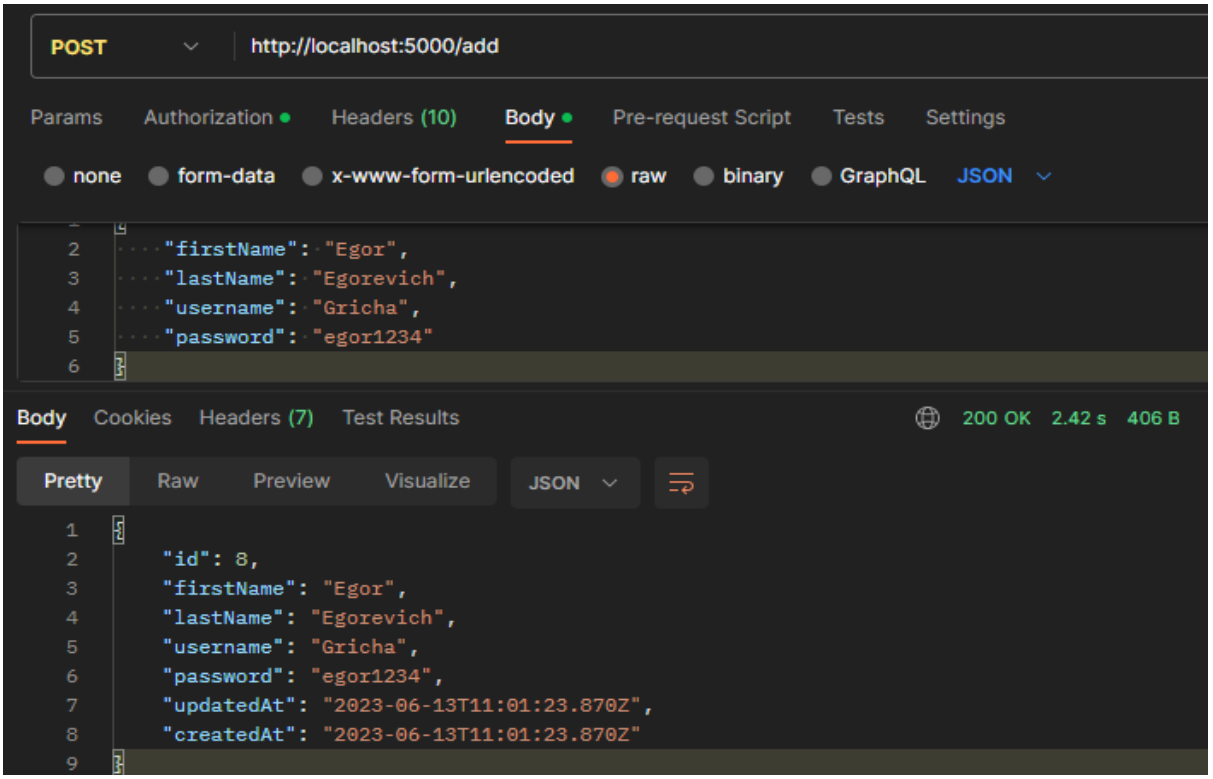
```
'use strict';
const {
  Model
} = require('sequelize');
module.exports = (sequelize, DataTypes) => {
  class User extends Model {

    static associate(models) {

    }
  }
  User.init({
    firstName: DataTypes.STRING,
    lastName: DataTypes.STRING,
    username: DataTypes.STRING,
    password: DataTypes.STRING
  }, {
    sequelize,
    modelName: 'User',
  });
  return User;
};
```

routes

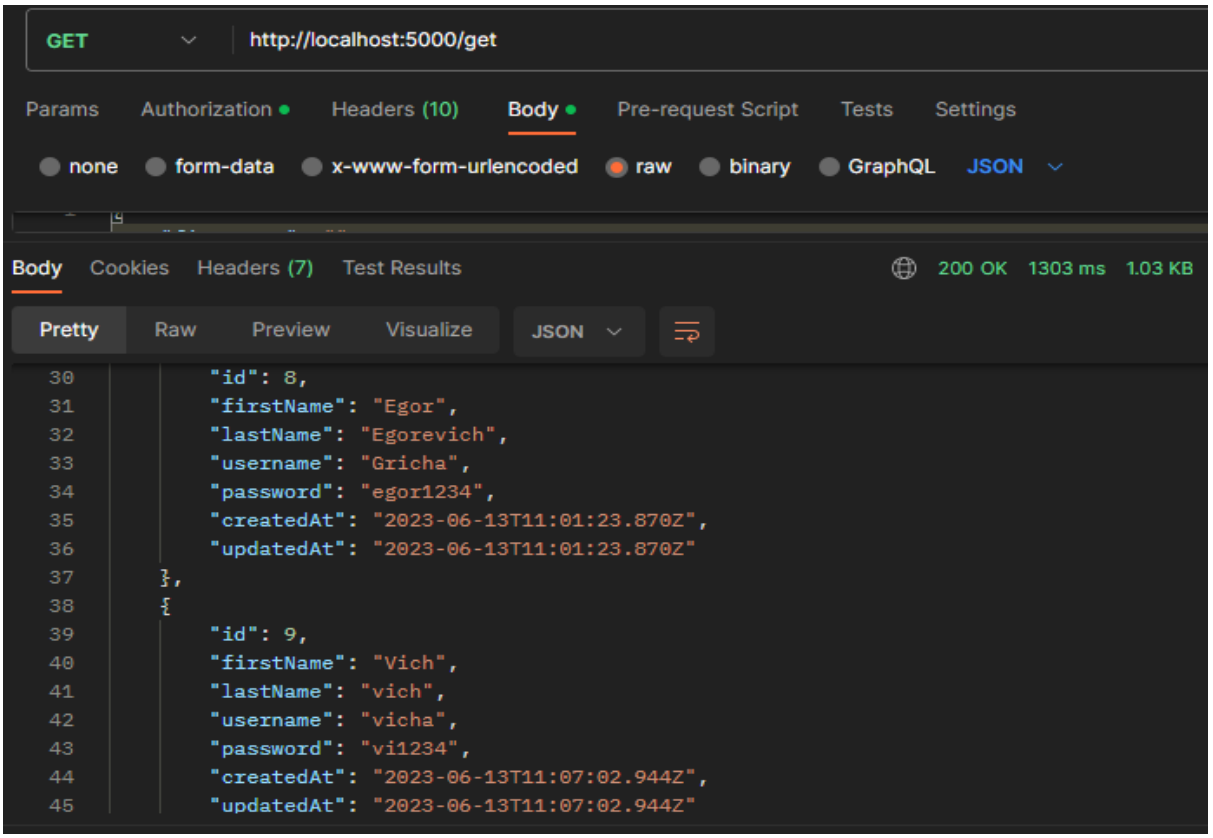
```
const express = require('express')
const db = require('./models')
const bodyParser = require("body-parser");

const app = express()
const port = 5000

app.use(bodyParser.json())

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

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

app.post('/add', async (req, res) => {
    const user = await db.User.create(req.body)
    res.send(user.toJSON())
})

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

```

```
app.get('/users/id/:id', async (req, res) => {
    const user = await db.User.findByPk(req.params.id)
    if (user) {
        res.send(user.toJSON())
    } else {
        res.status(404).send({'msg': 'user not found'})
    }

})

app.get('/users/username/:username', async (req, res) => {
    const user = await db.User.findOne({
        where: {
            username: req.params.username
        }
    })
    if (user) {
        res.send(user.toJSON())
    } else {
        res.status(404).send({'msg': 'user not found'})
    }

})

app.put('/update/:id', async (req, res) => {
    const user = await db.User.findByPk(req.params.id)
    if (user) {
        await db.User.update(req.body, {
            where: {
                id: req.params.id
            }
        })
        res.send(user.toJSON())
    } else {
        res.status(404).send({'msg': 'user not found'})
    }
})

app.delete('/delete/:id', async (req, res) => {
    const user = await db.User.findByPk(req.params.id)
    if (user) {
        user.destroy();
        res.sendStatus(200).send({'msg': 'user deleted'})
    }
    res.status(404).send({'msg': 'user not found'})
})

```

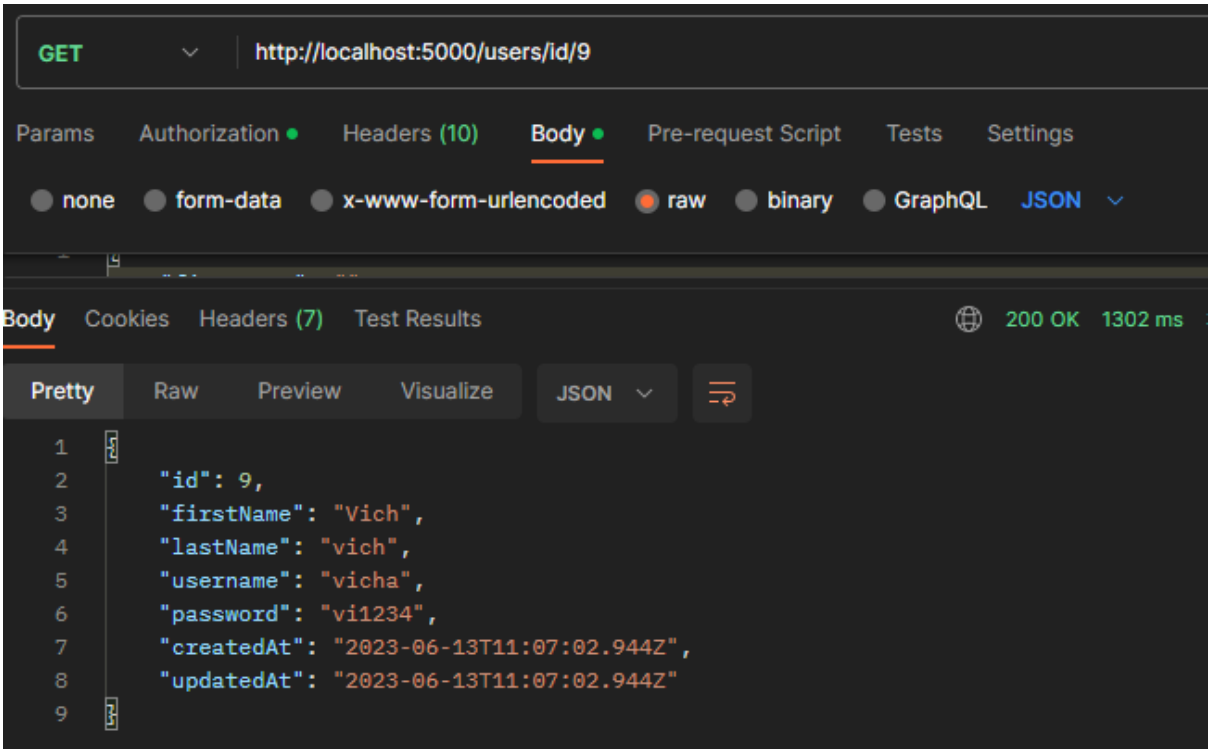
Add a new user



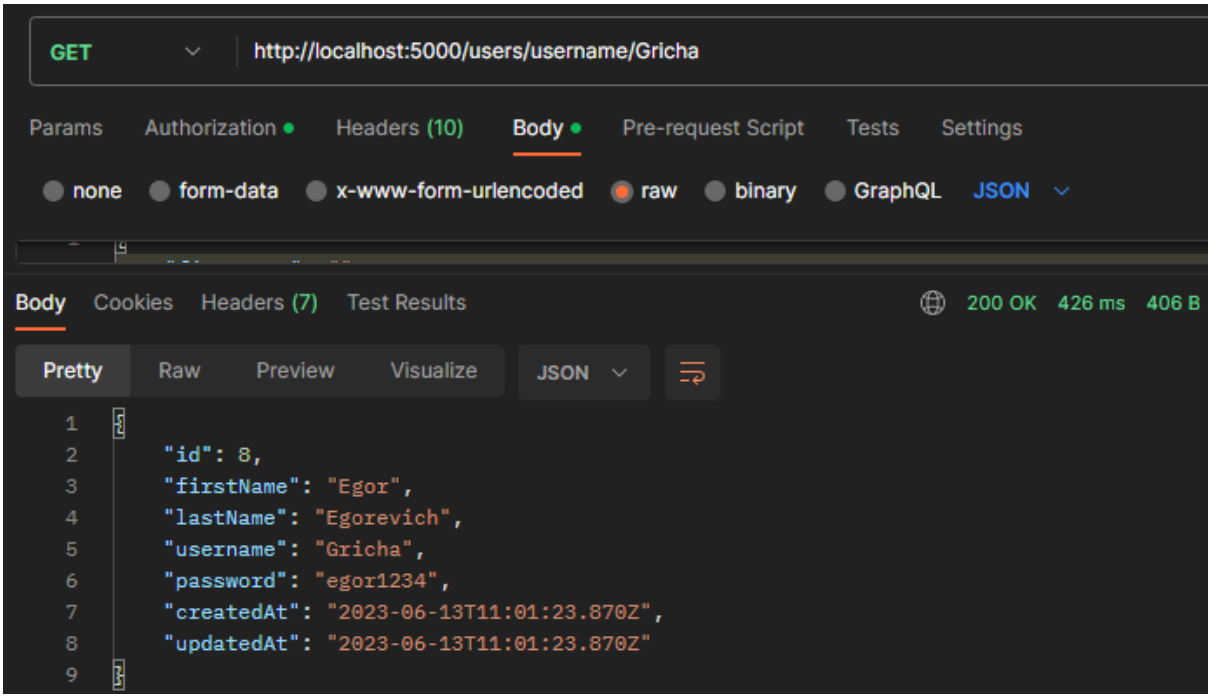
View all users



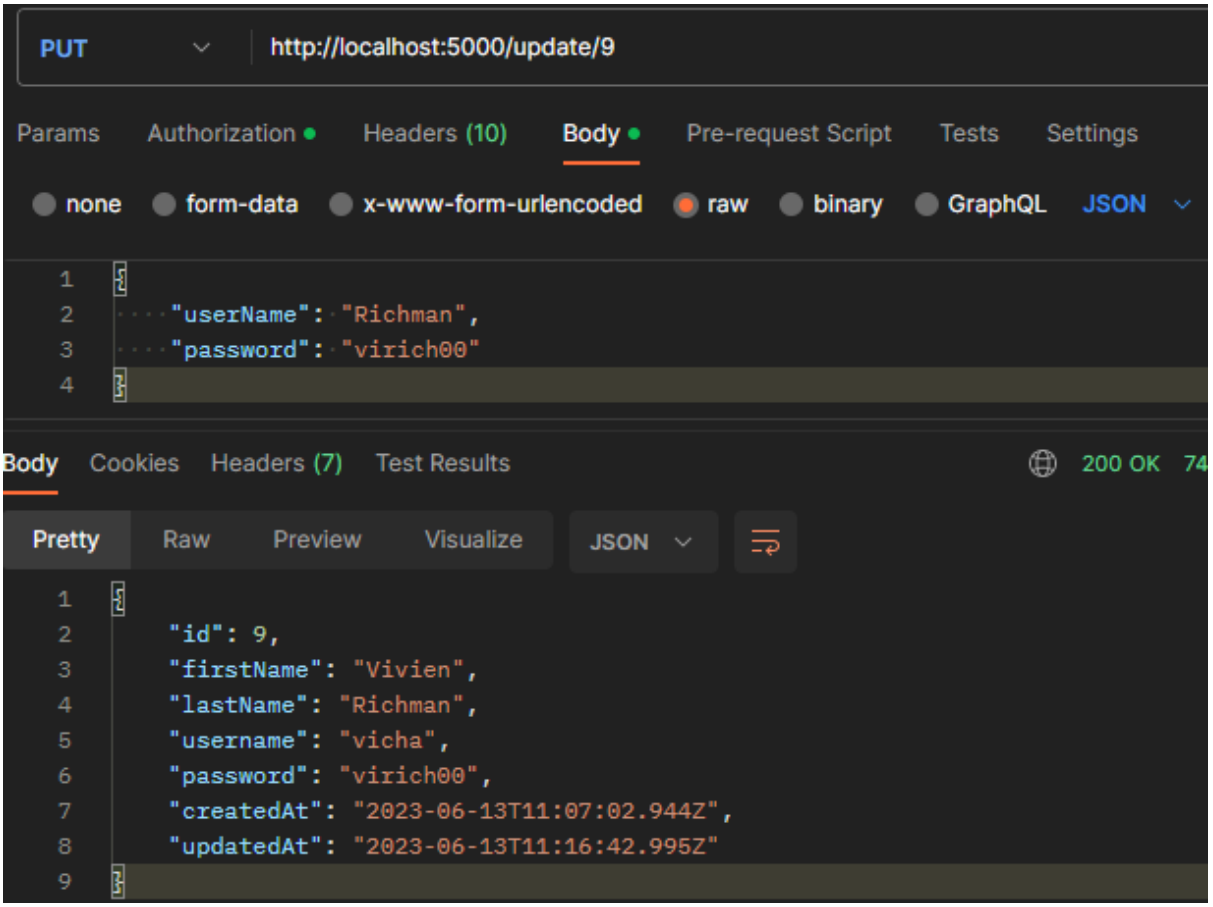
Get a user by id



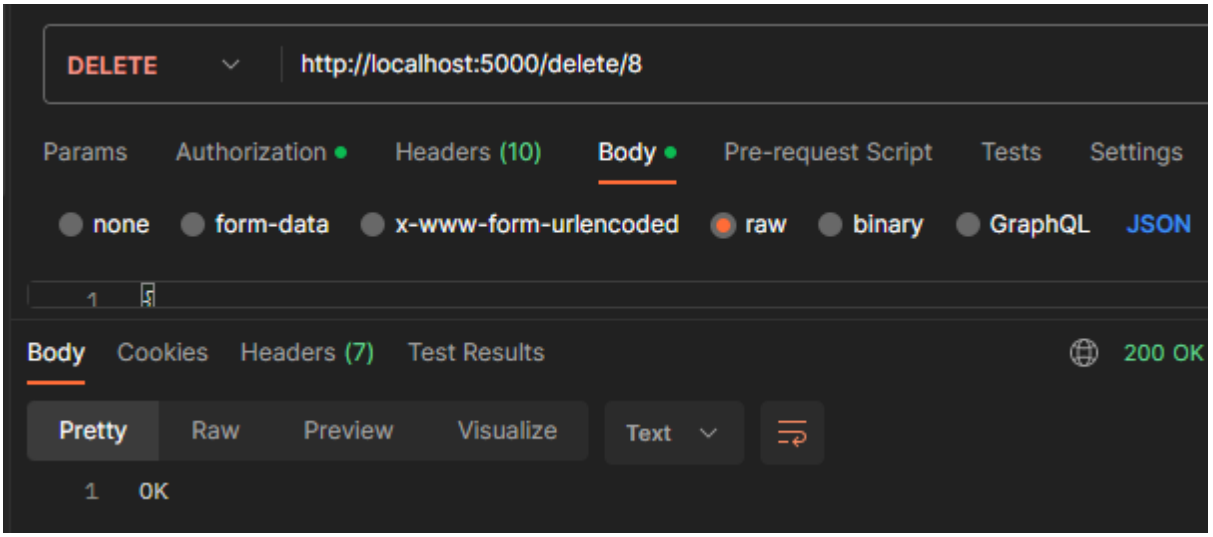
Get a user by username



Update user's data



Delete a user



Вывод:

При выполнения данной домашней работы, я научил работать с основами express и sequelize. Также, сумел создать скрипты и написал запросы для управления базой данных методом CRUD (create, read, update, delete).