САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа 2

Выполнил:

Таякин Даниил

Группа К33392

Проверил:
Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

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

Ход работы

1. Инициализируем проект

```
~/Documents/ITMO Projects/Sem-6/ITMO-ICT-Backend-2024/homeworks/K33392/Таякин_Даниил/hw2 git:(hw2) (15.16s)
npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (hw2)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /Users/taiakin/Documents/ITMO Projects/Sem-6/ITMO-ICT-Backend-2024/homeworks/K33
kage.json:

{
  "name": "hw2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes)
```

2. Устанавливаем зависимости

```
~/Documents/ITMO Projects/Sem-6/ITMO-ICT-Backend-2024/homeworks/K33392/Таякин_Даниил/hw2 git:(hw2)±1 (13.925s)
npm i express sequelize sqlite3 sequelize-cli
npm WARN deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs

added 316 packages, and audited 317 packages in 14s

40 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

3. Инициализируем sequelize

```
~/Documents/ITMO Projects/Sem-6/ITMO-ICT-Backend-2024/homeworks/K33392/Таякин_Даниил/hw2 git:(hw2)±1 (0.964s)
npx sequelize-cli init

Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.3]

Created "config/config.json"
Successfully created models folder at "/Users/taiakin/Documents/ITMO Projects/Sem-6/ITMO-ICT-Backe
Таякин_Даниил/hw2/models".
Successfully created migrations folder at "/Users/taiakin/Documents/ITMO Projects/Sem-6/ITMO-ICT-B
392/Таякин_Даниил/hw2/migrations".
Successfully created seeders folder at "/Users/taiakin/Documents/ITMO Projects/Sem-6/ITMO-ICT-Back
/Таякин_Даниил/hw2/seeders".
```

4. Генерируем модель при помощи sequelize

```
~/Documents/ITMO Projects/Sem-6/ITMO-ICT-Backend-2024/homeworks/K33392/Таякин_Даниил/hw2 git:(hw2)±1 (0.71s)
npx sequelize-cli model:generate --name User --attributes "username:string, email:string, password:string, firstName:string
, lastName:string, isAdmin:boolean"

Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.3]

New model was created at /Users/taiakin/Documents/ITMO Projects/Sem-6/ITMO-ICT-Backend-2024/homeworks/K33392/Таякин_Даниил/
hw2/models/user.js .
New migration was created at /Users/taiakin/Documents/ITMO Projects/Sem-6/ITMO-ICT-Backend-2024/homeworks/K33392/Таякин_Дан
иил/hw2/migrations/20240504061200-create-user.js .
```

5. Проводим миграцию

```
~/Documents/ITMO Projects/Sem-6/ITMO-ICT-Backend-2024/homeworks/K33392/Таякин_Даниил/hw2 git:(hw2)±1 (0.987s)
npx sequelize-cli db:migrate

Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.3]

Loaded configuration file "config/config.json".
Using environment "development".
== 20240504061200-create-user: migrating =======
== 20240504061200-create-user: migrated (0.007s)
```

6. Создаем в папке проекта index.js и прописываем эндпоинты
7. Создаем endpoint для получения всех пользователей

```
// Get all users
app.get('/users', async (req, res) => {
  try {
    res.json(await db.User.findAll())
  } catch (error) {
    res.status(500).json({ error: error.message })
  }
})
```

8. Создаем endpoint для создания пользователей

```
// Create a new user
app.post('/users', async (req, res) => {
  try {
    const user = await db.User.create(req.body)
    res.status(200).json(user)
  } catch (error) {
    res.status(400).json({ error: error.message })
  }
})
```

9. Создаем endpoint для получения пользователей по id

```
// Get user by ID
app.get('/users/:id', async (req, res) => {
  try {
    const user = await db.User.findByPk(req.params.id)
    if (!user) {
      res.status(404).json({ error: 'User not found' })
    } else {
      res.json(user)
    }
  } catch (error) {
    res.status(500).json({ error: error.message })
  }
})
```

10. Создаем endpoint для обновления данных пользователей по id

```
// Update user by ID
app.patch('/users/:id', async (req, res) => {
  try {
    const user = await db.User.findByPk(req.params.id)
    if (!user) {
      res.status(404).json({ error: 'User not found' })
    } else {
      await user.update(req.body)
      res.json(user)
    }
  } catch (error) {
    res.status(500).json({ error: error.message })
  }
})
```

11. Создаем endpoint для удаления пользователей

```
// Delete user by ID
app.delete('/users/:id', async (req, res) => {
  try {
    const user = await db.User.findByPk(req.params.id)
    if (!user) {
      res.status(404).json({ error: 'User not found' })
    } else {
      await user.destroy()
      res.status(200).send()
    }
  } catch (error) {
    res.status(500).json({ error: error.message })
  }
})
```

Вывод

В данной домашней работе было выполнено создание базового HTTP-сервера, обрабатывающий CRUD операции, с использованием express и sequilize библиотек, а также использованием sqlite базы данных.