САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа №2

Знакомство с ORM Sequelize

Выполнила:
Зайцева А. А.
Группа К33402

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

1. Продумать свою собственную модель пользователя

Разработанная модель пользователя включает следующие поля:

- **id** (integer, autoincrement, pk) – создается автоматически;
- **createdAt** (date) – создается автоматически;
- **updatedAt** (date) – создается автоматически;
- **email** (string, unique, not null);
- **password** (string, not null);
- **firstName** (string);
- **lastName** (string);
- **dob** (date).

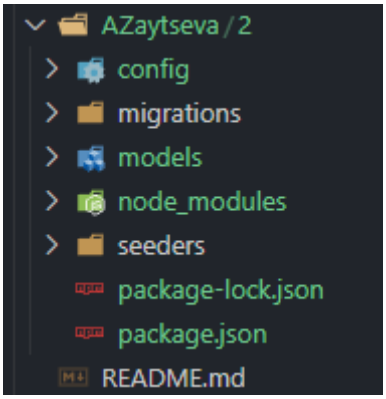
2. Реализовать набор из CRUD-методов для работы с пользователями средствами Express + Sequelize

Сначала инициализируем проект и установим все необходимые зависимости:

```
$ npm init
$ npm install --save sequelize sqlite3 express
$ npm install --save-dev sequelize-cli nodemon
```

Теперь инициализируем проект с помощью sequelize-cli:

```
$ npx sequelize init
```

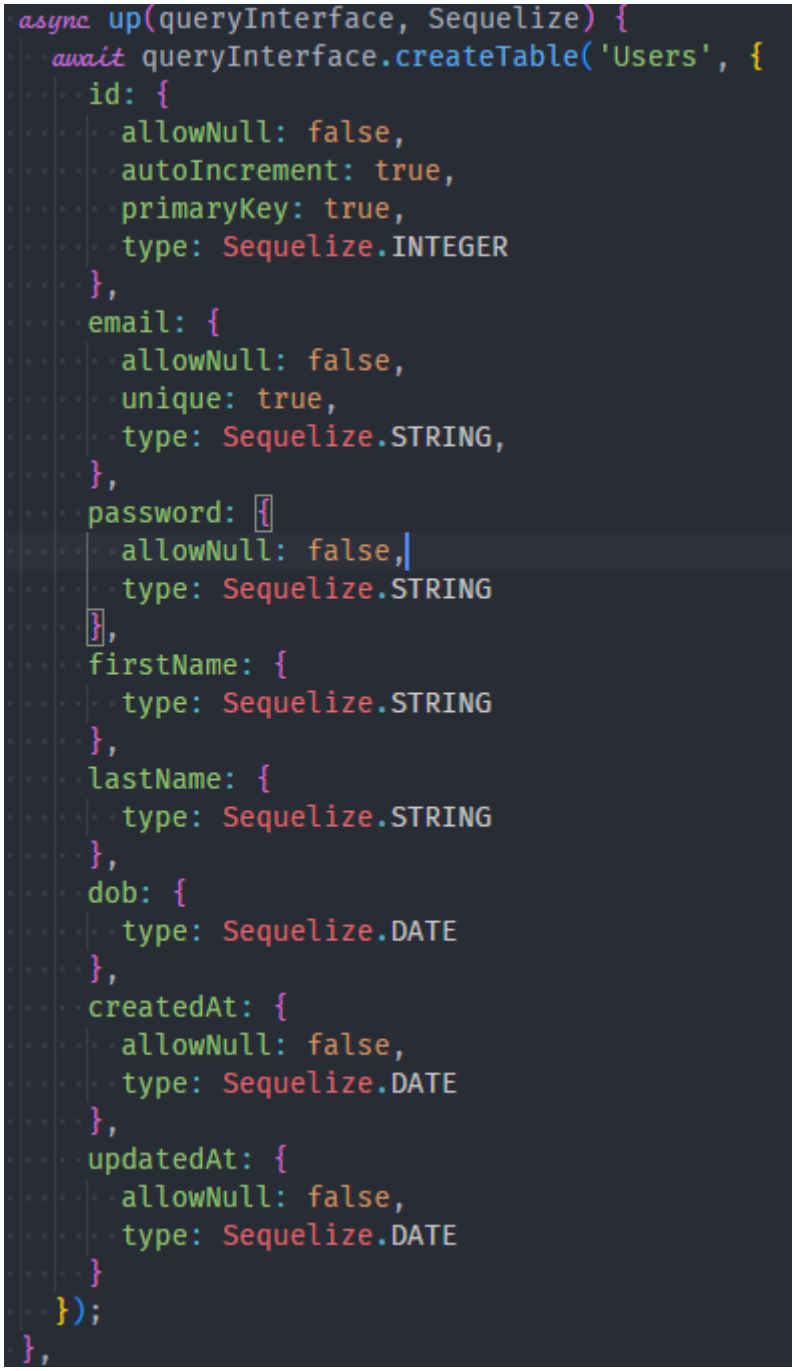


Теперь создадим модель и миграцию пользователя с помощью sequelize-cli:

```
$ npx sequelize-cli model:generate --name User --
attributes
email:string,password:string,firstName:string,lastName:str
ing,dob:date
```

Теперь отредактируем получившиеся файлы модели и миграции в соответствии с разработанной нами моделью пользователя:

```
User.init({
  email: {
    type: DataTypes.STRING,
    allowNull: false,
    unique: true,
  },
  password: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  firstName: DataTypes.STRING,
  lastName: DataTypes.STRING,
  dob: DataTypes.DATE
}, {
```



Запустим миграцию с помощью команды:

```
$ npx sequelize-cli migrate
```

```
~/p/s/6/ITMO-ICT-Backend-2023/h/K33402/AZaytseva/2 | hw2 ?1 | npx sequelize db:migrate

Sequelize CLI [Node: 18.12.1, CLI: 6.6.0, ORM: 6.29.3]

Loaded configuration file "config/config.json".
Using environment "development".
== 20230312170940-create-user: migrating =======
== 20230312170940-create-user: migrated (0.084s)
```

Для тестирования создадим seed с тестовыми пользователями базы данных:

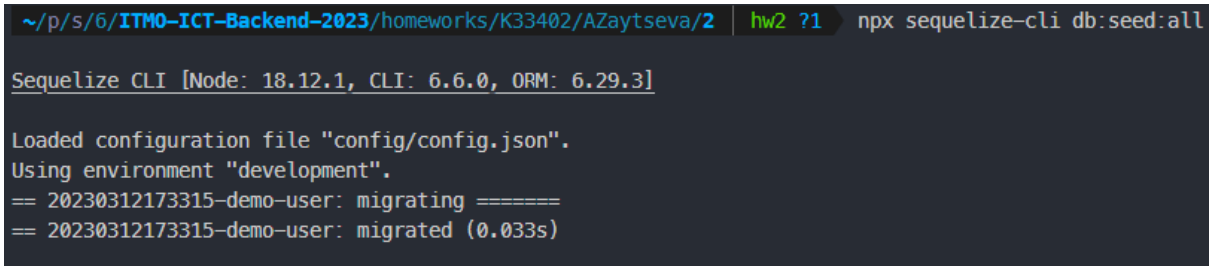
```
$ npx sequelize-cli seed:generate --name demo-user
```

homeworks > K33402 > AZaytseva > 2 > seeders > JS 20230312173315-demo-user.js > <unknov

```
'use strict';

/** @type {import('sequelize-cli').Migration} */
module.exports = {
  async up (queryInterface, Sequelize) {
    return queryInterface.bulkInsert('Users', [
      {
        firstName: 'First',
        lastName: 'User',
        email: 'first@example.com',
        password: 'Testtest123',
        dob: new Date('2002-02-02'),
        createdAt: new Date(),
        updatedAt: new Date()
      },
      {
        firstName: 'Second',
        lastName: 'User',
        email: 'second@example.com',
        password: 'Testtest123',
        dob: new Date('2002-02-02'),
        createdAt: new Date(),
        updatedAt: new Date()
      },
    ]);
  },

  async down (queryInterface, Sequelize) {
    return queryInterface.bulkDelete('Users', null, {});
  }
};

```

Запустим seed командой:

```
$ npx sequelize-cli db:seed:all
```



Теперь напишем набор CRUD-методов для работы с пользователем.
Были реализованы следующие методы:

- GET /users[?email]
- GET /users/:id
- POST /users
- PUT /users/:id
- DELTE /users
- DELTE /users/:id

Полный код реализованных методов доступен в директории практической работы в файле index.js.

3. Написать запрос для получения пользователя по id/email

Для получения по id:

```
curl -X GET http://localhost:3000/users/2
```

Для получения по email:

```
curl -X GET \
'http://localhost:3000/users?email=third%40example.com'
```

Вывод

В ходе второй практической работы я научилась работать с orm npm пакетом sequelize. Была разработана модель пользователя, подготовлена и запущена миграция средствами sequelize-cli. С помощью фреймворка express был разработан набор CRUD методов для работы с сущностью пользователя. Набор реализованных методов был протестирован в программе Postman. Там же были сгенерированы curl запросы.