САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет по домашней работе
Знакомство с ORM Sequelize

Выполнил:
Малышенко А. Р.
Группа К33392
Проверил:
Добряков Д. И.

Санкт-Петербург
2024 г.

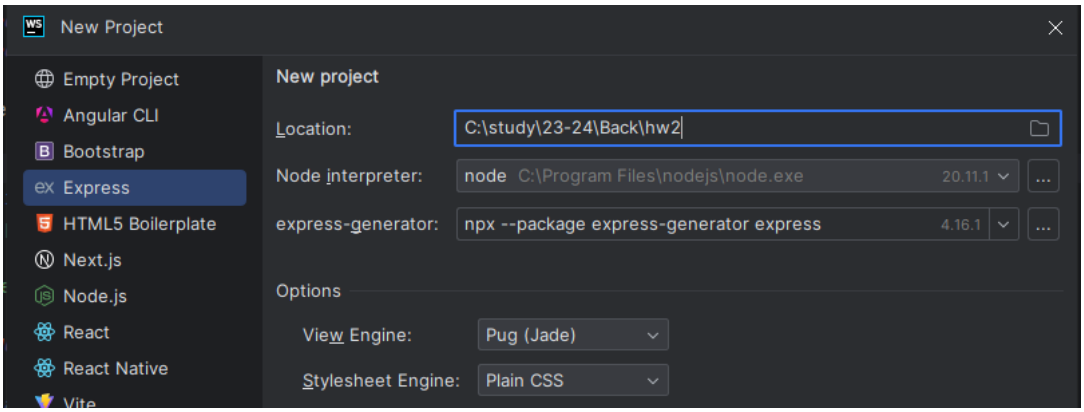
Содержание отчета

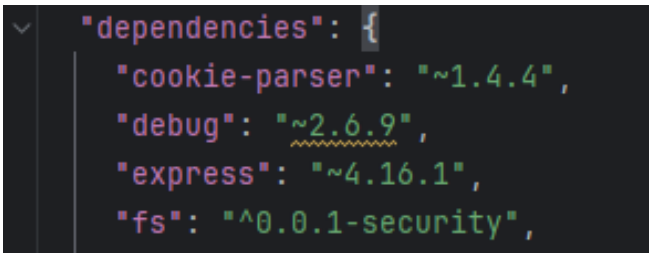
1) Задача

- 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.

2) Создание проекта

Проект создается в ide WebStorm, благодаря чему express устанавливается в проект по умолчанию

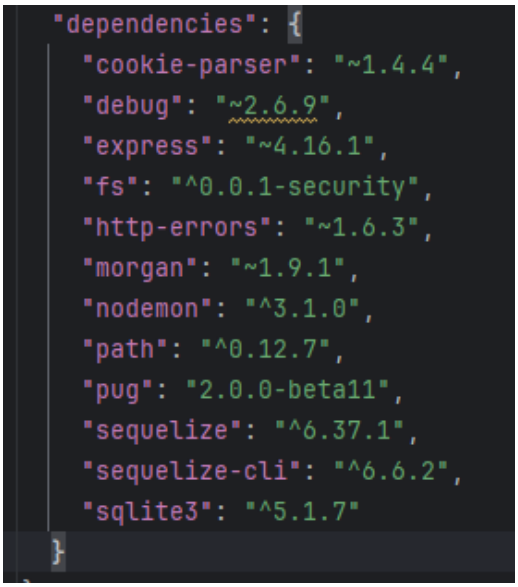




3) Установка зависимостей

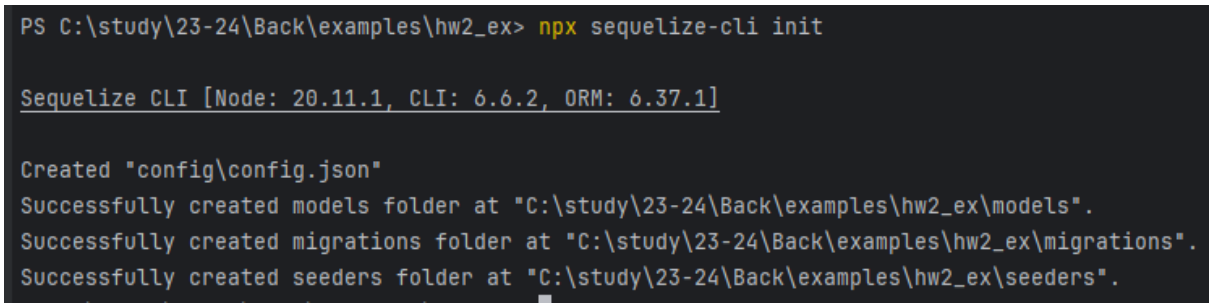
Устанавливаем необходимые зависимости:

> *npm install sqlite3 sequelize sequelize-cli*



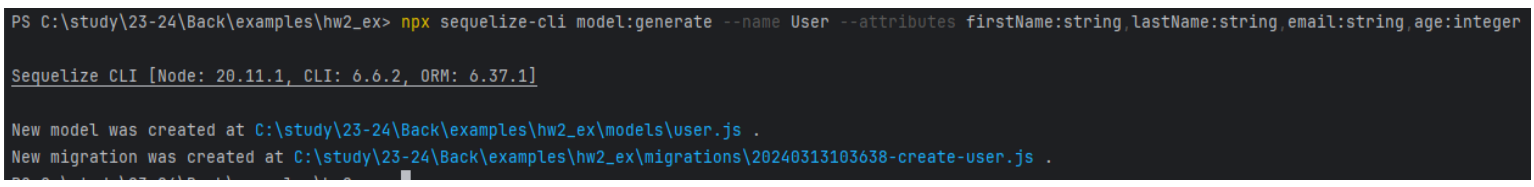
4) Инициализация

> *npx sequelize-cli init*



5) Создание модели User

> *npx sequelize-cli model:generate --name User --attributes firstName:string,lastName:string,email:string,age:integer*



Для миграции используем следующую команду:

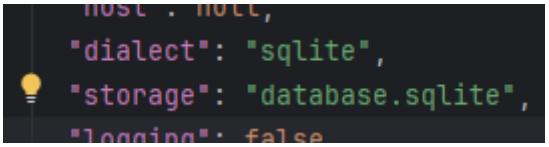
> *npx sequelize db:migrate*

```
PS C:\study\23-24\Back\examples\hw2_ex> npx sequelize db:migrate

Sequelize CLI [Node: 20.11.1, CLI: 6.6.2, ORM: 6.37.1]

Loaded configuration file "config\config.json".
Using environment "development".
== 20240313103638-create-user: migrating =======
== 20240313103638-create-user: migrated (0.019s)
```

Так же перед миграцией надо не забыть изменить тип используемой ORM в config.json на sqlite:



6) Создание эндпоинтов

Создание пользователя

```
// create new user
app.post('/users', async (req, res) => {
    await db.User.create(req.body).then(() => {
        res.send({"msg": 'user created'})
    })
})
```

Вывод пользователей всех пользователей

```
// get users
app.get('/users', async (req, res) => {
    const users = await db.User.findAll()
    if (!users) {
        return res.send({"msg": "users is not found"})
    }

    return res.send(users)
})
```

Вывод пользователя по его id

```
// get user by id
app.get('/users/:id', async (req, res) => {
    const user = await db.User.findByPk(req.params.id)
    if (user) {
        return res.send(user)
    }

    return res.send({"msg": "user is not found"})
})
```

Удаление пользователя по его id

```
// delete user by id
app.delete('/users/:id', async (req, res) => {
    const user = await db.User.findByPk(req.params.id)
    if (user) {
        await user.destroy()
        return res.send({"msg": "user deleted"})
    }

    return res.send({"msg": "user is not found"})
})
```