САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа № 1

Выполнил:

Казанков Илья К33402

Проверил: Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

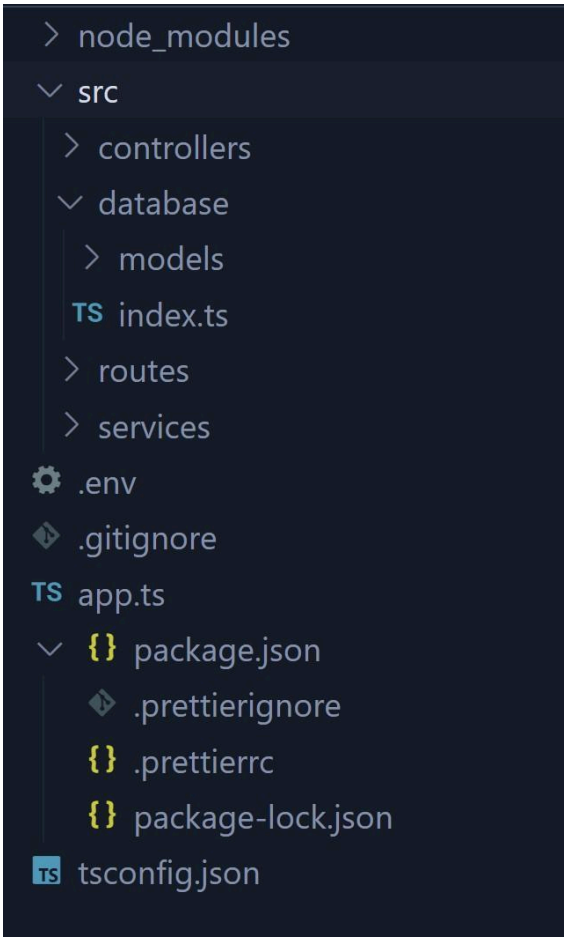
Нужно написать свой boilerplate на express + sequelize / TypeORM + typescript.
Должно быть явное разделение на:

- модели
- контроллеры
- роуты
- сервисы для работы с моделями (реализуем паттерн “репозиторий”)

Ход работы

В начале работы инициализировал репозиторий. Добавил нужные зависимости, настроил Prettier для удобного форматирования кода, поработал с tsconfig.json файлом, подробно настроив TypeScript.

Определил структуру файлов. Проект запускается через app.ts файл, а остальной код содержится в директории src, и разделён по принципам MVC.



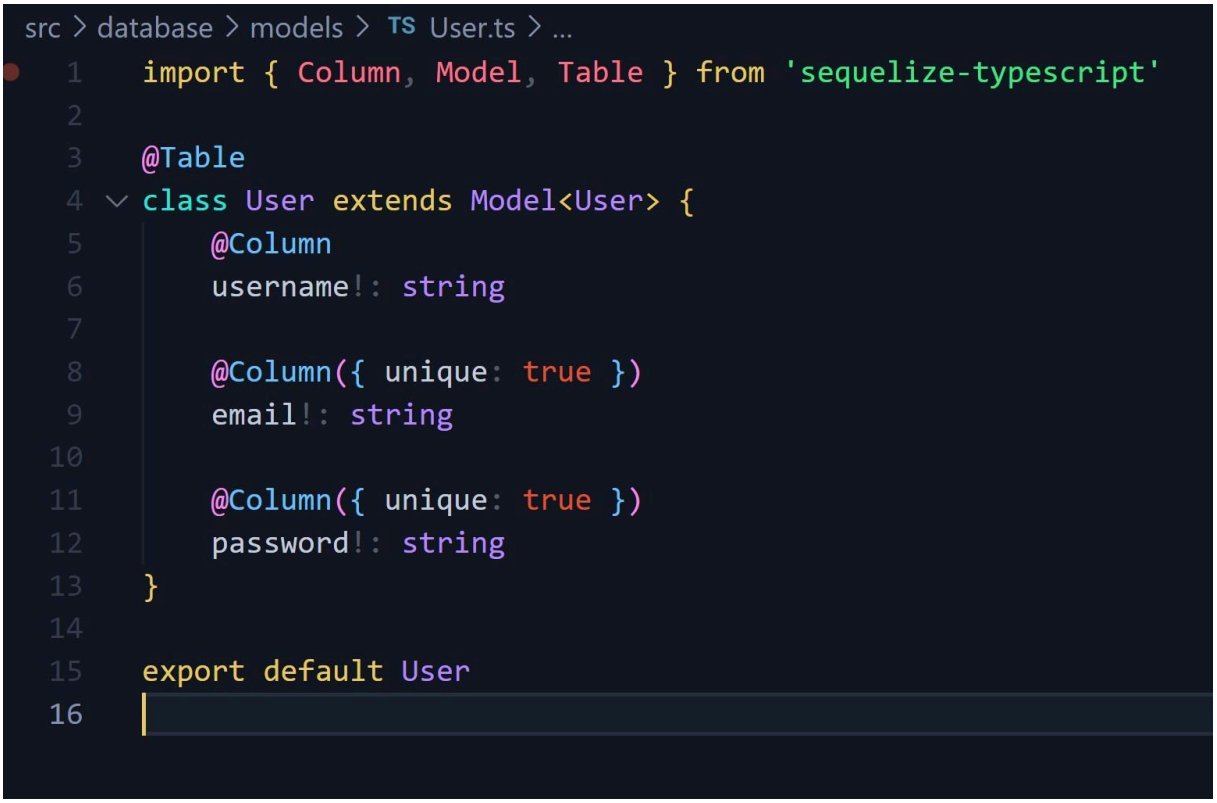
В главном файле подключаем Express JS и Sequelize. С помощью bodyParser активируем работу энд поинтов.

```ts
import bodyParser from 'body-parser'
import express from 'express'
import sequelize from 'src/database'

const PORT = Number(process.env.PORT || 5000)

const app = express()

app.use(bodyParser.json())

app.listen(PORT, () => {
    sequelize // to not delete after compilation
    console.log(`Server is running on port ${PORT}`)
})
```

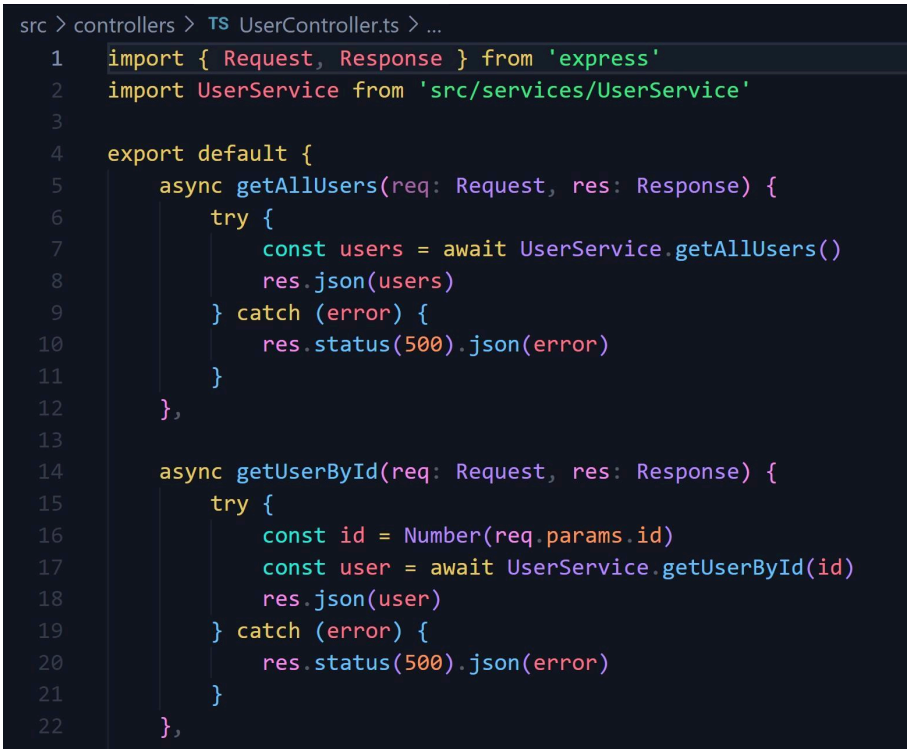
При подключении базы данных используем переменные окружения.

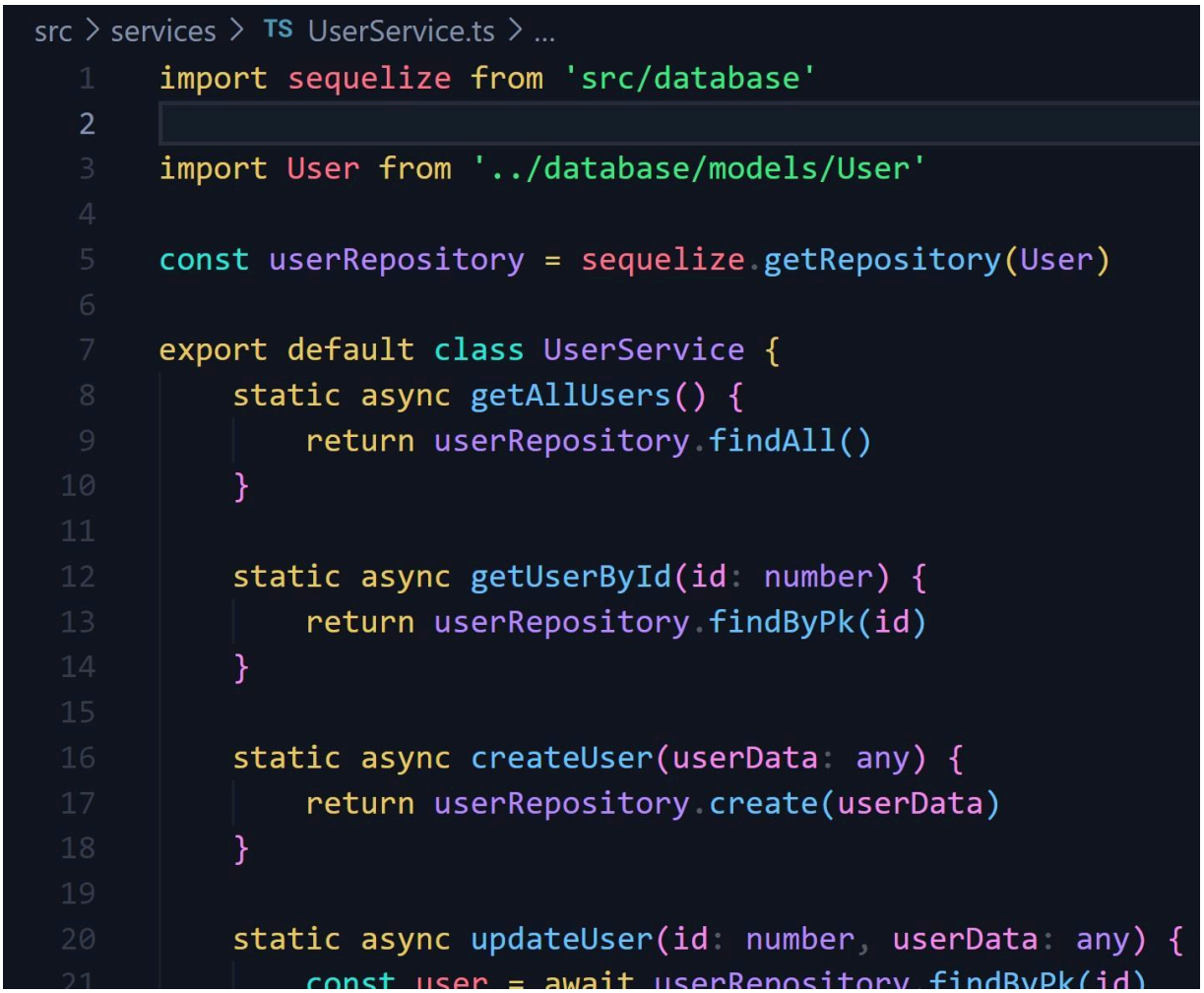
```ts
import { Sequelize } from 'sequelize-typescript'

import User from './models/User'

const sequelize = new Sequelize({
    database: process.env.DB_NAME || 'database123',
    username: process.env.DB_USER || 'username123',
    password: process.env.DB_PASSWORD || 'password123',
    dialect: 'postgres',
    host: process.env.DB_HOST || 'localhost',
    port: Number(process.env.DB_PORT || '5432'),
    repositoryMode: true,
    logging: console.log,
})

sequelize.addModels([User])

sequelize.sync().then(() => {
    console.log('sync sequelize')
})
```

Создаём модель пользователя, используя sequelize-typescript и декораторы



Ошибки обрабатываем внутри Controllers, а бизнес логика находится внутри Services и соответствует паттерну Repository.





Вывод

В процессе работы создали boilerplate репозиторий с ExpressJs и sequelize, получили навыки составления структуры серверного проекта