САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

ЛР 3 - миграция написанного API на микросервисную архитектуру

Выполнил:
Сергеев Виктор
K3341

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

- выделить самостоятельные модули в вашем приложении;
- провести разделение своего API на микросервисы (минимум, их должно быть 3).

Ход работы

В приложении были выделены следующие самостоятельные модули:

- сервис работы с пользователем - информация о нём + авторизация
- сервис работы с рецептами - хранение рецептов, ингредиентов, тегов и шагов
- социальный сервис - лайки и комментарии

Был создан новый проект, в котором будет содержаться миросервисное приложение. В проекте было создано 3 директории для каждого сервиса и создано окружения со всеми библиотеками. Также, каждый сервис должен собственную базу данных, поэтому в был создан докер-композ, в котором поднимается 3 контейнера с postgres и в каждой создаётся своя база данных.

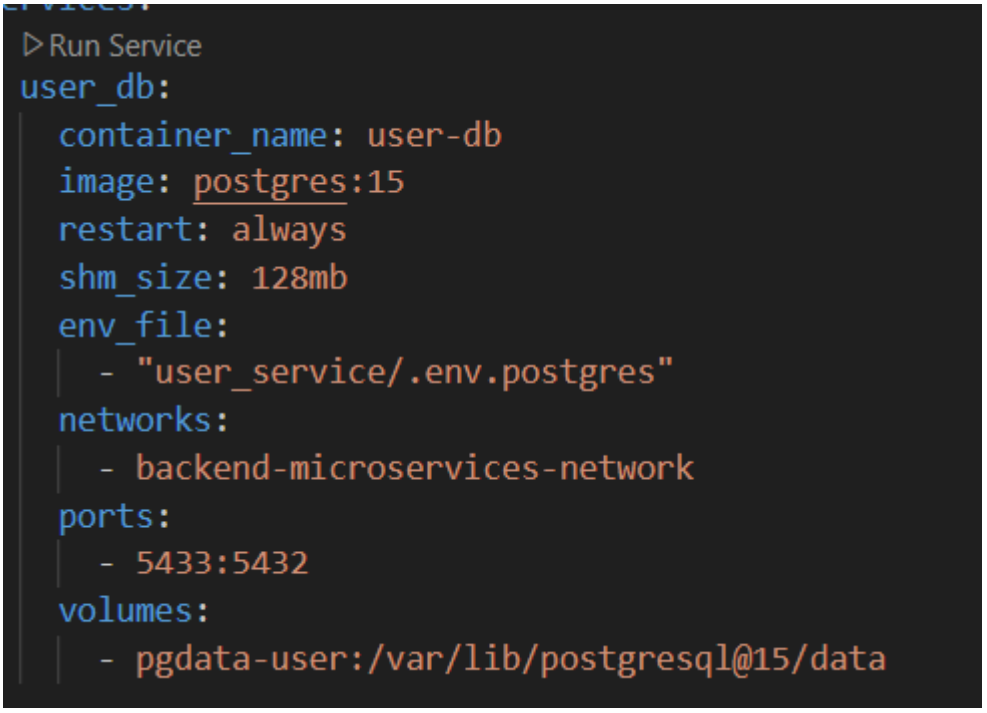

Рисунок 1 - пример контейнера с базой данных для сервиса пользователей

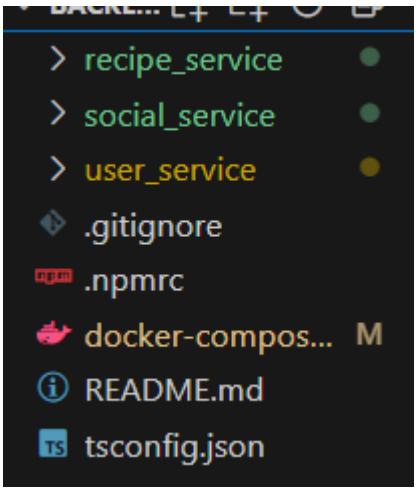

Рисунок 2 - структура микросервисного проекта

Поскольку теперь приложение разделено, как и хранилище данных, требуется поменять модели и дто, чтобы модели не хранили другие модели, о которых в рамках своего сервиса они не знают. Вместо этого в моделях будут хранится ключи на эти модели в явном виде.

social_service > src > models > TS Comment.ts > Comment

```
1   import { Entity, PrimaryGeneratedColumn, Column } from "typeorm";
2
3   @Entity()
4   export class Comment {
5
6       @PrimaryGeneratedColumn()
7       id: number
8
9       @Column({ type: "integer" })
10      user_id: number
11
12      @Column({ type: "integer" })
13      recipe_id: number
14
15      @Column({type: "varchar", length: 200})
16      comment: string
17
18      @Column({type: "timestamp", update: false, default: () => "CURRENT_TIMESTAMP"})
19      created_at: Date
20  }
```

Рисунок 3 - модель комментария без зависимостей

Остальные модули остались без изменений - проект просто был разделён на несколько частей, но сохранил практически всю свою функциональность.

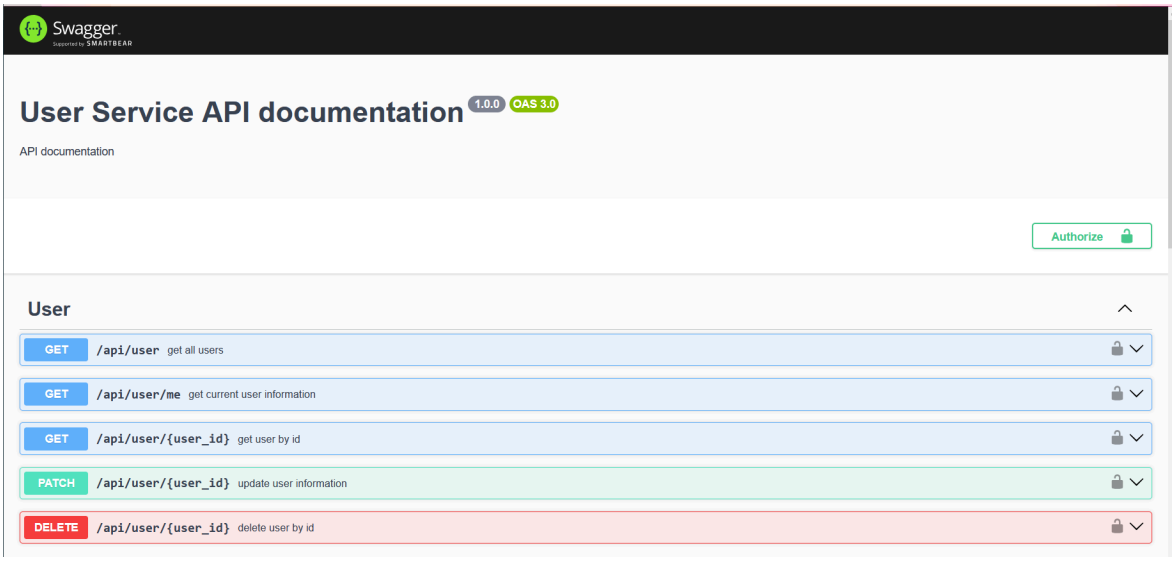

Рисунок 4 - сваггер сервиса пользователей

Вывод

В процессе работы ранее написанное приложение было разделено на самостоятельные модули, которые были разделены в отдельные микросервисы. Каждый микросервис представляет собой отдельное приложение на node JS.