

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа № 3
“Микросервисы”

Выполнил:

Коротин А.М.

К33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

Необходимо реализовать отдельный микросервис, выполняющий какую-либо содержательную функцию из всего арсенала функций вашего приложения.

Ход работы

В отдельный микросервис была вынесена логика аутентификации. Во время проектирования архитектуры рассматривались следующие варианты реализации работы с БД:

- 1) Database-per-service подход — каждый сервис оперирует своей собственной БД,
- 2) Shared-database подход — каждый сервис оперирует одной общей базой данных.

Выбор пал на использование 2ого подхода, поскольку он упрощает поддержание консистентности данных в системе.

Стоит отметить, что реализация 1ого подхода также допустима, поскольку в сервисе аутентификации настроена система доменных событий (например, на регистрацию нового пользователя). С их помощью можно было асинхронно передавать информацию остальным сервисам системы (например, с использованием брокера сообщений). Однако, такая система выглядит достаточно монструозно и излишне в контексте данной лабораторной работы, поэтому и был выбран более простой Shared-database подход.

На рисунке 1 приведена файловая структура проекта с использованием микросервисной архитектуры.

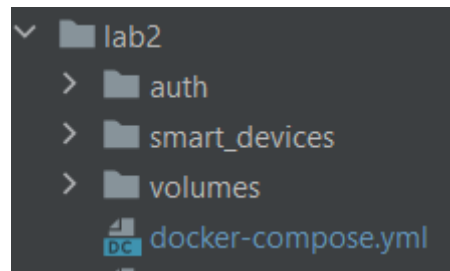


Рисунок 1 — Файловая структура

В микросервисе `smart_devices` была объявлена репликационная `sequelize`-модель аккаунта пользователя, предназначенная только для чтения (рисунок 2).

```
1+ usages  Alexey Korotin
@Table( options: {tableName: 'Accounts'})
export class AccountModel extends Model {
  @PrimaryKey
  @Column(DataType.UUID)
  declare id: string

  @Unique
  @Column
  declare email: string

  @Column(DataType.ENUM(...Object.keys(AccountRole)))
  declare role: AccountRole

  @Column(DataType.ENUM(...Object.keys(AccountStatus)))
  declare status: AccountStatus

  @Column
  declare password: string
}
```

Рисунок 2 — Репликационная модель аккаунта

Именно эта модель будет использоваться для дальнейших отношений с другими моделями сервиса.

Вывод

В ходе выполнения лабораторной работы были изучены основы микросервисной архитектуры. Также было произведено разбиение ранее написанного приложения на микросервисы, взаимодействующие между собой