

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

**Отчет**

**Лабораторная работа №6**

**Выполнил:**

**Борисова Элина**

**Группа  
К3441**

**Проверил:  
Добряков Д. И.**

**Санкт-Петербург**

**2025 г.**

## Задача

Подключение очередей сообщений для настройки межсервисного взаимодействия

## Ход работы

В каждый ранее созданный микро сервис был добавлен докер файл. Также в корне проекта есть docker - compose файл.

`services:`

```
auth-service:
  build:
    context: ./microservices dockerfile:
      auth-service/Dockerfile

  ports:
    - "3000:3000"

  env_file:
    - ./microservices/auth-service/.env

  depends_on:
    - auth-db

  network:
    s:
      - app-network

recipe-service:
  build:
    context: ./microservices recipe-service dockerfile:
      recipe-service/Dockerfile
```

```
dockerfile: recipe-service/Dockerfile

ports:
  - "3001:3001"

env_file:
  - ./microservices/recipe-service/.e
nv depends_on:
  - recipe-
db networks:
  - app-network

social-service:

build:
  context: ./microservices

  dockerfile: social-service/Dockerfile

ports:
  - "3002:3002"

env_file:
  - ./microservices/social-service/.e
nv depends_on:
  - social-
db networks:
  - app-network
```

```
auth-db:  
  image: postgres:14  
  env_file:  
    - ./microservices/auth-service/.env  
  ports:  
    - "5433:5432"  
  volumes:  
    - auth-db-data:/var/lib/postgresql/data  
  networks:  
    - app-network  
  
recipe-db:  
  image: postgres:14  
  env_file:  
    - ./microservices/recipe-service/.env  
  ports:  
    - "5434:5432"  
  volumes:  
    - recipe-db-data:/var/lib/postgresql/data  
  networks:  
    - app-network  
  
social-db:
```

В docker-compose.yml был добавлен сервис RabbitMQ

```
version: "3.9"
services:
rabitmq:
image: "rabitmq:3-management"
container_name: rabbitmq
ports:
- "5672:5672"      # основной порт для обмена
сообщениями
- "15672:15672"    # веб-панель управления
environment:
RABBITMQ_DEFAULT_USER: user
RABBITMQ_DEFAULT_PASS: password
networks:
- app-network

networks:
app-network:
driver: bridge
```

Каждый микросервис запускается на своем порту.

Запуск проекта:

docker-compose up --build

Итог:

## auth-service

<http://localhost:3000/docs/>

Recipe Service API 1.0.0 OAS 3.0

Authorize

### Auth

POST /auth/register Регистрация пользователя

POST /auth/login Авторизация пользователя

### User

GET /user Список пользователей

POST /user Создать пользователя

GET /user/{id} Пользователь

PUT /user/{id} Обновить пользователя

DELETE /user/{id} Удалить пользователя

## recipe-service

<http://localhost:3001/docs/>

Authorize

### DishType

GET /dish-type/ Список типов блюд

POST /dish-type/ Создать тип блюда

GET /dish-type/{id} Тип блюда

PUT /dish-type/{id} Обновить тип блюда

DELETE /dish-type/{id} Удалить тип блюда

### Ingredient

GET /ingredient/ Список ингредиентов

POST /ingredient/ Создание ингредиента

GET /ingredient/{id} Ингредиент

PUT /ingredient/{id} Обновление ингредиента

DELETE /ingredient/{id} Удаление ингредиента

### RecipeDifficulty

## social-service

<http://localhost:3002/docs/>

## Comment

**POST** /comment/{recipeId} Создать комментарий

**GET** /comment/{recipeId} Комментарии к рецепту

**PUT** /comment/{id} Обновить комментарий

**DELETE** /comment/{id} Удалить комментарий

## Like

**POST** /like Создать лайк

**GET** /like/{id} Лайк

**DELETE** /like/{id} Удалить лайк

**GET** /like/recipe/{recipeId} Лайки для рецепта

**GET** /like/user/{userId} Лайки пользователя

## SavedRecipe

**GET** /saved-recipe Список сохраненных рецептов

**POST** /saved-recipe Сохранить рецепт

**GET** /saved-recipe/{id} Сохраненный рецепт

## Вывод

В данной работе были реализованы докер файлы для каждого микросервиса и файл docker - compose.