

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Практическая/Лабораторная работа

Выполнил:

Габов Михаил

К3340

Проверил:

Добряков Д. И.

Санкт-Петербург

2022 г.

## 1. Выделение самостоятельных модулей

В ходе анализа исходного монолитного приложения были выделены следующие логически независимые домены, которые стали основой для микросервисов:

- **Аутентификация и пользователи (auth-users):** Этот модуль отвечает за все, что связано с пользователями: регистрация, аутентификация, получение информации о пользователях и управление подписками.
- **Рецепты (recipes):** Этот модуль управляет созданием, поиском, обновлением и удалением рецептов и их ингредиентов.
- **Взаимодействия (interactions):** Этот модуль обрабатывает действия пользователей, связанные с рецептами, такие как комментарии, лайки и добавление в избранное.

## 2. Разделение API на микросервисы

На основе выделенных модулей было создано три отдельных микросервиса, для оркестрации запуска используется concurrently.

### Сервис auth-users

- **Ответственность:** Управление пользователями, аутентификация (JWT), подписки.
- **Порт:** 3001
- **Модели:** User, Follow.
- **Контроллеры:** UsersController, FollowController.
- **База данных:** users\_db.

### Сервис recipes

- **Ответственность:** Управление рецептами, ингредиентами, шагами приготовления.
- **Порт:** 3002
- **Модели:** Recipe, Ingredient, RecipeIngredient, RecipeStep.
- **Контроллеры:** RecipeController, IngredientController.
- **База данных:** recipes\_db.

## Сервис interactions

- **Ответственность:** Управление комментариями, лайками, избранным.
- **Порт:** 3003
- **Модели:** Comment, Like, Favorite.
- **Контроллеры:** CommentController, LikeController, FavoriteController.
- **База данных:** interactions\_db.

## 3. Настройка сетевого взаимодействия

Для обеспечения взаимодействия между клиентом и микросервисами, а также для централизации некоторых сквозных задач, был внедрен **API Gateway**.

### API Gateway (api-gateway)

- **Ответственность:** Единая точка входа, маршрутизация запросов, централизованная аутентификация.
- **Порт:** 3000
- **Технологии:** Express, express-http-proxy.

### Маршрутизация

API Gateway перенаправляет входящие запросы на соответствующие микросервисы на основе URL-пути:

- `http://localhost:3000/users/*` -> `http://localhost:3001/users/*`
- `http://localhost:3000/follows/*` -> `http://localhost:3001/follows/*`
- `http://localhost:3000/recipes/*` -> `http://localhost:3002/recipes/*`
- `http://localhost:3000/ingredients/*` -> `http://localhost:3002/ingredients/*`
- `http://localhost:3000/comments/*` -> `http://localhost:3003/comments/*`
- `http://localhost:3000/likes/*` -> `http://localhost:3003/likes/*`
- `http://localhost:3000/favorites/*` -> `http://localhost:3003/favorites/*`

### Аутентификация

API Gateway выполняет централизованную аутентификацию. Он извлекает JWT из заголовка Authorization, верифицирует его и, в случае успеха, добавляет ID пользователя в заголовок x-user-id. Этот заголовок затем

проксируется в нижестоящие сервисы, которые могут доверять этой информации и использовать `userId` для своих внутренних нужд, избавляясь от необходимости дублировать логику проверки токена.

## **Вывод**

В результате работы монолитная архитектура была преобразована в микросервисную.