

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

**Отчет**

**Лабораторная работа №5**

**Выполнил:**

**Зотов Михаил**

**К3441**

**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);
- настроить сетевое взаимодействие между микросервисами.

## Ход работы

### 1. Выделение микросервисов.

В проекте реализованы три сервиса:

- **auth-service** (порт 7000) — регистрация, авторизация, проверка токенов ([/auth/verify](#)).
- **recipes-service** (порт 7001) — CRUD для рецептов, категории, ингредиенты, комментарии.
- **subscription-service** (порт 7002) — подписки пользователей, сохранённые рецепты, лайки.

### 2. Использование Nginx как API Gateway.

Вместо Node-gateway для маршрутизации используется **Nginx**, который:

- проверяет токен перед проксированием запросов;
- проксирует запросы к нужным сервисам;
- добавляет заголовки **x-user-id**, **x-user-name** для идентификации пользователя в downstream-сервисах.

### 3. Конфигурация Nginx.

Основные правила в **nginx.conf**:

```
server {
```

```
listen 8080;

# Все запросы к recipes-service
location / {
    auth_request /check-token;
    auth_request_set $user_id
$upstream_http_x_user_id;
    auth_request_set $user_name
$upstream_http_x_user_name;
    proxy_set_header x-user-id $user_id;
    proxy_set_header x-user-name $user_name;

    proxy_pass http://recipe-service:7001;
    proxy_request_buffering on;
    proxy_buffering off;
}

# Проверка токена через auth-service
location /check-token {
    internal;
    proxy_pass
http://auth-service:7000/auth/verify;
```

```
proxy_method GET;

proxy_set_header Content-Length 0;

}

# Доступ напрямую к auth-service

location /auth-service {

    rewrite ^/auth-service/(.*) /$1 break;

    proxy_pass http://auth-service:7000;

}

# subscription-service с авторизацией

location /subscription-service {

    rewrite ^/subscription-service/(.*) /$1
break;

    auth_request /check-token;

    auth_request_set $user_id
$upstream_http_x_user_id;

    auth_request_set $user_name
$upstream_http_x_user_name;

    proxy_set_header x-user-id $user_id;

    proxy_set_header x-user-name $user_name;
}
```

```
    proxy_pass http://subscription-service:7002;  
    proxy_request_buffering on;  
    proxy_buffering off;  
}  
}
```

#### 4. Механизм работы.

- Все запросы к `recipes-service` и `subscription-service` проходят проверку через `/check-token`.
- `auth-service` возвращает данные о пользователе (`id`, `name`) в заголовках.
- Эти заголовки прокидываются дальше в сервисы, чтобы они могли привязывать действия к конкретному пользователю.
- `auth-service` доступен напрямую через префикс `/auth-service`, чтобы можно было регистрироваться и логиниться без проверки токена.

#### Вывод

В ходе работы:

- API разделено на три микросервиса: `auth-service`, `recipes-service`, `subscription-service`.

- Для маршрутизации и проверки авторизации настроен **Nginx в роли API Gateway**.
- Межсервисное взаимодействие организовано через HTTP с передачей пользовательских данных в заголовках.