

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №5

Выполнил:

Бархатова Наталья

Группа К3439

**Проверил:
Добряков Д. И.**

Санкт-Петербург

2026 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

Было выделено 4 микросервиса:

- 1) Микросервис аутентификации
- 2) Микросервис пользователей (User, UserDetails)
- 3) Микросервис тренировок (Workout, TrainingPlan, Progress)
- 4) Микросервис блога (Post)

Основное взаимодействие между микросервисами заключается в аутентификации. Теперь в middleware каждого микросервиса происходит обращение к сервису аутентификации.

```
import { ExpressMiddlewareInterface, ForbiddenError,
UnauthorizedError } from 'routing-controllers';
import { AuthClient } from '../services/AuthClient';

export class AuthMiddleware implements ExpressMiddlewareInterface {
    private authClient = new AuthClient();

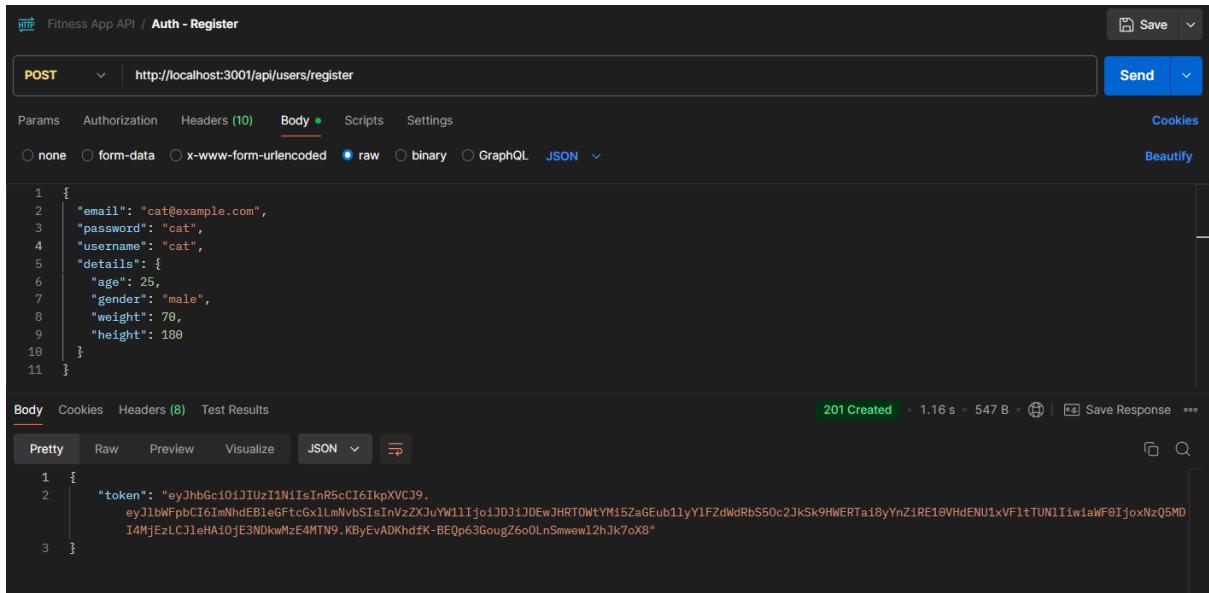
    async use(req: any, res: any, next: Function): Promise<any> {
        const authHeader = req.headers['authorization'];
        const token = authHeader && authHeader.split(' ')[1];

        if (!token) {
            throw new UnauthorizedError('No token provided');
        }

        try {
            const user = await this.authClient.validateToken(token);
            req.user = user;
            next();
        } catch (error: any) {
            throw new ForbiddenError('Invalid token: ' + (error?.message
            || 'Unknown error'));
        }
    }
}
```

Для сервиса пользователей также были настроены запросы в сервис аутентификации для получения токена при регистрации и логине.

Для каждого микросервиса кроме микросервиса аутентификации была создана своя бд, параметры подключения вынесены в .env



The screenshot shows a Postman interface with the following details:

- Method: POST
- URL: <http://localhost:3001/api/users/register>
- Body tab selected
- Content Type: raw (JSON)
- JSON content:

```
1 {
2   "email": "cat@example.com",
3   "password": "cat",
4   "username": "cat",
5   "details": {
6     "age": 25,
7     "gender": "male",
8     "weight": 70,
9     "height": 180
10  }
11 }
```
- Response status: 201 Created
- Response body:

```
1 {
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
3       eyJlbWFpbC16ImNhEB1eGftGx1lmNv0SIsInVzZXJuYW1IjoiJDJiJDEwJHRT0WtyM15ZaGEub1lyY1FzdWdRbS5Oc2JkSk9HWERTa18yYnZ1RE19VHdENU1xVFltTUNlIiwiWF0IjoxNzQ5MD
4       T4MjEzL01eHA10jf3NDkwMzE4MTN9.KByEvADKhfIK-BEOp63GougZ6oLnSmwewl2hJk7oX8"
```

Вывод

Приложение было успешно разделено на четыре микросервиса с отдельными базами данных и .env-конфигурациями. Аутентификация централизована: все сервисы используют middleware для проверки JWT через обращение к auth-сервису.