

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэкенд разработка

Отчет

Лабораторная работа №3

Выполнил:

Фамилия Имя
Тишалович Леонид

Группа
К33392

Проверил:
Добряков Д. И.

Санкт-Петербург

2024 г.

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

Ход выполнения:

В ходе выполнения лабораторной работы, проект приобрёл следующую структуру:

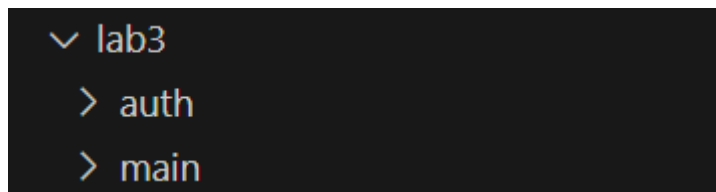


Рисунок 1 - структура проекта lab3

Как видно из структуры проекта авторизация и регистрация были вынесены в отдельный микросервис. Вся логика авторизации и регистрации была вынесена в микросервис (логика была показана в отчете к прошлой лабораторной работе). Ниже показаны изменения в основном приложении.

```
import express from "express";
import UserController from "../controllers/UserControllers";

const router = express.Router();

router.post("/login", UserController.login);
router.post("/register", UserController.register);

export default router;
```

Рисунок 2 - authRoutes.ts

```

export default class UserController {
  static async login(req: Request, res: Response) {
    try {
      const data = await AuthService.login(req.body.email, req.body.password);
      res.json(data);
    } catch (error) {
      res
        .status(error.response?.status || 500)
        .json({
          message: error.response?.data?.message || "Authentication failed",
        });
    }
  }

  static async register(req: Request, res: Response) {
    try {
      const data = await AuthService.register(
        req.body.email,
        req.body.name,
        req.body.password
      );
      res.json(data);
    } catch (error) {
      res
        .status(error.response?.status || 500)
        .json({
          message: error.response?.data?.message || "Registration failed",
        });
    }
  }
}

```

Рисунок 3 - часть UserController.ts

В AuthService, как видно, отправляется запрос на запущенный микросервис с помощью axios.

```

import axios from "axios";

const AUTH_SERVICE_URL = "http://localhost:4000/auth";

export default class AuthService {
  static async login(email: string, password: string) {
    const response = await axios.post(`${AUTH_SERVICE_URL}/login`, {
      email,
      password,
    });
    return response.data;
  }

  static async register(email: string, name: string, password: string) {
    const response = await axios.post(`${AUTH_SERVICE_URL}/register`, {
      email,
      name,
      password,
    });
    return response.data;
  }
}

```

Рисунок 4 - AuthService.ts

Вывод: в ходе лабораторной работы было изучено понятие микросервиса в бэкенд проекте и получены практические навыки некоторые функции выносить из основного проекта в микросервис для лучшей расширяемости.