

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа № 3

Выполнил:

Пырков Владислав

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

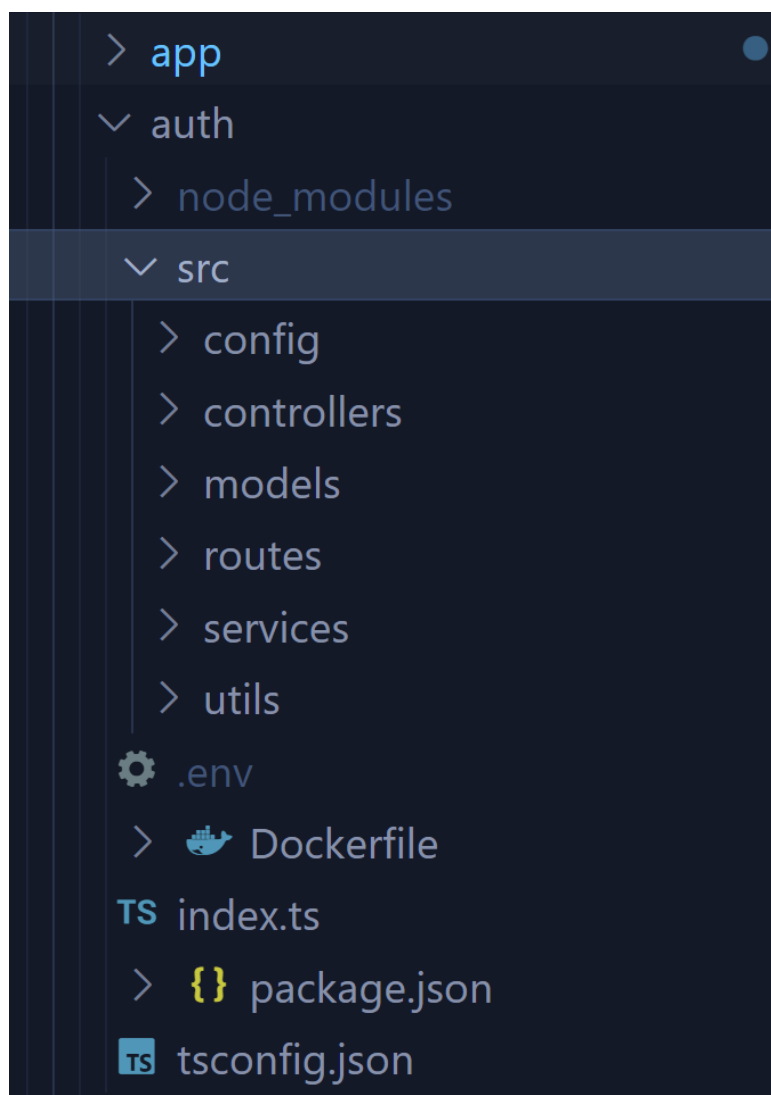
2024 г.

Задача

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

Ход работы

Для данной работы в качестве микросервера я выбрал функционал авторизации и регистрации. Все связанные с этим части приложения были вынесены в отдельную папку, и запускаются отдельно от остального кода.



В этой части приложения осуществляется работа с пользователем, профилем, а также токенами, которые выдаются при авторизации.

```

labs > K33402 > Пырков_Владислав > lab1 > auth > src > routes > TS AuthRoutes.ts > [x] userRou
1  import express from 'express'
2
3  import AuthController from '../controllers/AuthController'
4  import ProfileController from '../controllers/ProfileController'
5  import UserController from '../controllers/UserController'
6
7  const userRoutes = express.Router()
8
9  userRoutes.post('/reg', AuthController.reg)
10 userRoutes.post('/login', AuthController.login)
11 userRoutes.post('/check_token', AuthController.verify)
12
13 userRoutes.get('/profile', ProfileController.getProfileById)
14 userRoutes.post('/profile', ProfileController.createOrUpdateProfi
15
16 userRoutes.get('/user/:id', UserController.getUserById)
17 userRoutes.post('/user', UserController.createUser)
18 userRoutes.delete('/user/:id', UserController.deleteUser)
19
20 export default userRoutes
21

```

Токены выдаются с помощью JsonWebToken и находятся в заголовках всех последующих запросов

```

36  static async login(email: string, password: string) {
37      try {
38          const user = await userRepository.findOne({ where: { email } })
39          if (!user) {
40              return serviceHandleError({ message: 'Некорретный адрес почты' })
41          }
42
43          const passwordMatch = await user.checkPassword(password)
44          if (!passwordMatch) {
45              return serviceHandleError({ message: 'Некорректный пароль' })
46          }
47
48          const key = process.env.JWT_SECRET_KEY || ''
49          const token = jwt.sign({ userId: user.id }, key, {
50              expiresIn: '1h',
51          })
52
53          return token
54      } catch (error) {
55          return serviceHandleError({

```

Также в директории микросервиса авторизации и регистрации содержатся свои переменные окружения, нужные для взаимодействия с другими частями приложения.

```
labs > K33402 > Пырков_Владислав > lab1 > auth > ⚙ .env
1   PORT=8000
2   DB_NAME=postgres
3   DB_USERNAME=postgres
4   DB_PASSWORD=toraha01
```

Стоит отметить, что на данный момент наличие нескольких сепарированных частей приложения затрудняет его запуск. Этот аспект будет улучшен в следующей лабораторной работе с помощью Docker-compose.

Вывод

В процессе работы вынесли часть функционала приложения в микросервис, тем самым разделив его логику, и научились работать с приложениями, состоящими из нескольких отдельных частей.