

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа № 3

Выполнил:

Казанков Илья К33402

Проверил: Добряков Д. И.

Санкт-Петербург

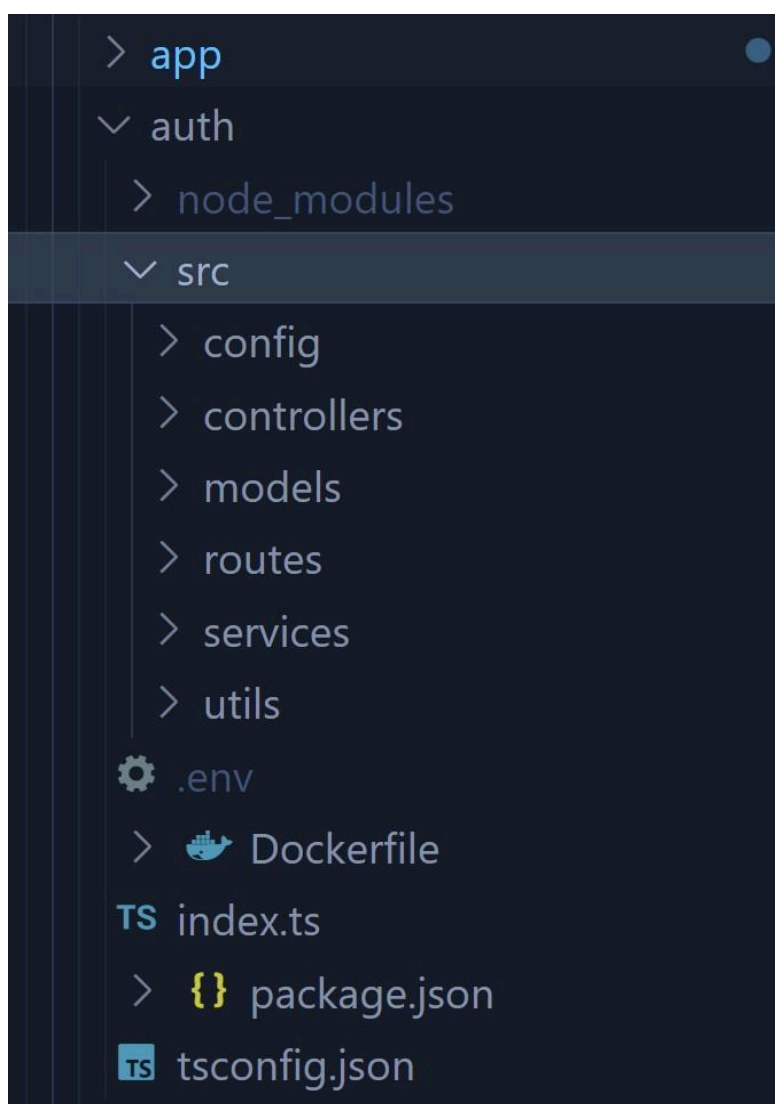
2024 г.

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.

```
app.post('/products', authMiddleware, adminMiddleware, productController.create);
app.put('/products/:id', authMiddleware, adminMiddleware, productController.update);
app.get('/products/search', productController.search);
app.post('/cart', authMiddleware, cartController.addToCart);
app.get('/cart', authMiddleware, cartController.getCart);
```

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

```
export const authMiddleware = (req: Request, res: Response, next: NextFunction): Response<any, Record<string, any>> | void => {
  const token = req.headers['authorization']?.split(' ')[1];

  if (!token) {
    return res.status(401).json({ error: 'No token provided' });
  }

  jwt.verify(token, process.env.JWT_SECRET || 'SECRET_KEY', (err: VerifyErrors | null, decoded: any) => {
    if (err) {
      return res.status(401).json({ error: 'Invalid token' });
    }

    (req as any).user = decoded;
    next();
  });
};
```

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

```
product-service > .env
1  PORT=8000
2  DB_NAME=postgres
3  DB_USERNAME=postgres
4  DB_PASSWORD=admin
5  DB_HOST=localhost
```

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.