## САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа №4: Документирование АРІ

Выполнил: Жаров Александр К33402

> Проверил: Добряков Д. И.

Санкт-Петербург

## Задача

1) Сделать документацию API с помощью Swagger

## Ход работы

Для создания документации я решил использовать swagger, который внедрял в проект на nestjs.

1) Добавил swagger в зависимости

```
"devDependencies": {
    "@nestjs/cli": "^9.0.0",
    "@nestjs/schematics": "^9.0.0",
    "@nestjs/swagger": "^7.3.1",
    "@nestjs/testing": "^9.0.0",
    "@types/bcryptjs": "^2.4.6",
    "@types/express": "^4.17.13",
    "@types/jest": "29.2.4",
    "@types/node": "18.11.18",
    "@types/passport-jwt": "^4.0.1",
```

2) В главном файле подключил его

```
async function bootstrap() {
    const config = app.get(ConfigService)
    const globalPrefix = 'api'

app.useGlobalPipes(new ValidationPipe())
app.setGlobalPrefix(globalPrefix)

const swaggerConfig = new DocumentBuilder()
    .setTitle('nestjs-prisma-boilerplate')
    .setDescription('Store application')
    .setVersion('1.0')
    .addTag('usth')
    .addTag('user')
    .addBearerAuth()
    .build()

const document = SwaggerModule.createDocument(app, swaggerConfig, {
    deepScanRoutes: true,
})

SwaggerModule.setup('api', app, document)

await app.listen(config.PORT)
Logger.log(
    *Application is running on: http://localhost:${config.PORT}/${globalPrefix}`
}

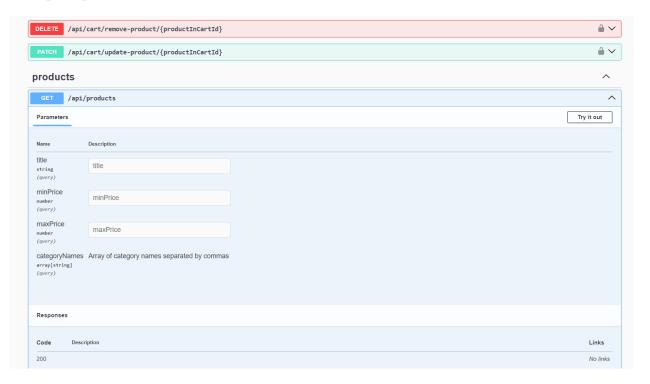
Application is running on: http://localhost:${config.PORT}/${globalPrefix}`
}

}
```

3) Добавил описание каждого эндпоинта

```
@ApiTags('products')
    @Controller('products')
14 ∨ export class ProductController {
      constructor(private readonly productService: ProductService) {}
      @ApiQuery({ name: 'title', required: false })
      @ApiQuery({ name: 'minPrice', required: false, type: 'number' })
      @ApiQuery({ name: 'maxPrice', required: false, type: 'number' })
      @ApiQuery({
        name: 'categoryNames',
        required: false,
        type: String,
        isArray: true,
        description: 'Array of category names separated by commas',
      @Get()
      async getAllProducts(
        @Query('title') title?: string,
        @Query('minPrice') minPrice?: string,
        @Query('maxPrice') maxPrice?: string,
        @Query('categoryNames') categoryNames?: string[]
       ): Promise<Product[]> {
        console.log(categoryNames)
```

4) Проверил получившуюся документацию



## Вывод

При выполнении работы я подключил к проекту swagger и научился внедрять документацию в бекенд, написанный на nestjs.