**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

# **Дисциплина:** Бэк-энд разработка

Отчет

Домашняя работа #3

Выполнил:

Игнатьев Алексей

К3340

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

**Задание:**

реализовать автодокументирование средствами swagger;

реализовать документацию API средствами Postman.

**Ход работы**

1. **Подключение Swagger в проект NestJs**

В рамках данного пункта была подклюена библиотека для генерации Swagger на основе DTO и методов, а также подключен плагин для автоматического распознавания DTO. Использовалась официальная документация Nest Js Swagger https://docs.nestjs.com/openapi/introduction

1. **Добавление комментариев к полям DTO**

Для некоторых DTO были добавлены комментарии о том, за что отвечают конкретные поля в DTO. Пример

import { Transform } from 'class-transformer';  
import { IsNumber } from 'class-validator';  
import { ApiProperty } from '@nestjs/swagger';  
export class CreateParticipantDto {  
 @Transform(({ value }) => parseInt(value, 10))  
 @IsNumber()  
 @ApiProperty({ description: 'ID аукциона' })  
 auction\_id: number;  
}

1. **Трансформация вложенных объектов**

Для корректного отображения вложенных DTO в Swagger был применён декоратор с трансформацией объектов. Пример

export class CreateBrandDto {  
 @IsNotEmpty()  
 @IsString()  
 @ApiProperty({ description: 'Название бренда' })  
 title: string; // Название бренда (Enum или строка)  
 @IsNotEmpty()  
 @IsArray()  
 @ValidateNested({ each: true })  
 @Type(() => LocalizationDto)  
 @ApiProperty({ description: 'Локализации', type: [LocalizationDto] })  
 locales: LocalizationDto[];  
}

**Вывод**

* В ходе выполнения домашней работы были настроены DTO, а также подключен Swagger из официального репозитория NestJs