

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Домашняя работа 3

Выполнила:

Фролова Кристина

Группа К3339

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

## **Задача**

- реализовать автодокументирование средствами swagger;
- реализовать документацию API средствами Postman.

## **Ход работы**

Так как для написания лабораторной работы 2 использовался фреймворк tsoa, то автодокументирование swagger происходило путём добавления аннотаций к контроллеру и его методам. На рисунке 1 представлен AdvertisementController, где можно посмотреть пример аннотаций, а на рисунках 2, 3, 4 – сам swagger.

На рисунке 5 представлена коллекция Postman

```

@Route("advertisements") Show usages
@Tags("Advertisement")
export class AdvertisementController extends Controller {

    @Get() Show usages
    @SuccessResponse("200", "Ok")
    @Security("jwt")
    public async getAll(): Promise<AdvertisementResponseDto[]> {
        const advertisements : Advertisement[] = await advertisementService.getAll();
        return advertisements.map(toAdvertisementResponseModel);
    }

    @Get("{advertisementId}") Show usages
    @SuccessResponse("200", "Ok")
    @Security("jwt")
    @Response<EntityNotFoundError>(404, "Entity not found")
    public async getById(@Path() advertisementId: number): Promise<AdvertisementResponseDto> {
        const advertisement : Advertisement = await advertisementService.getById(advertisementId);
        return toAdvertisementResponseModel(advertisement);
    }

    @Post() Show usages
    @SuccessResponse("201", "Created")
    @Security("jwt")
    public async create(
        @Body() body: CreateAdvertisementRequestDto
    ): Promise<AdvertisementResponseDto> {
        const advertisement : Advertisement = await advertisementService.create(createAdvertisementRequestToModel(body));
        return toAdvertisementResponseModel(advertisement);
    }

    @Put("{advertisementId}") Show usages
    @SuccessResponse("200", "Updated")
    @Security("jwt")
    @Response<EntityNotFoundError>(404, "Entity not found")
    public async update(
        @Path() advertisementId: number,
        @Body() body: UpdateAdvertisementRequestDto
    ): Promise<AdvertisementResponseDto> {
        const ad : Advertisement = await advertisementService.update(advertisementId, updateAdvertisementRequestToModel(body));
        return toAdvertisementResponseModel(ad);
    }

    @Delete("{advertisementId}") Show usages
    @SuccessResponse("204", "Deleted")
    @Security("jwt")
    public async delete(@Path() advertisementId: number): Promise<void> {
        await advertisementService.delete(advertisementId);
    }
}

```

Рисунок 1 – AdvertisementController

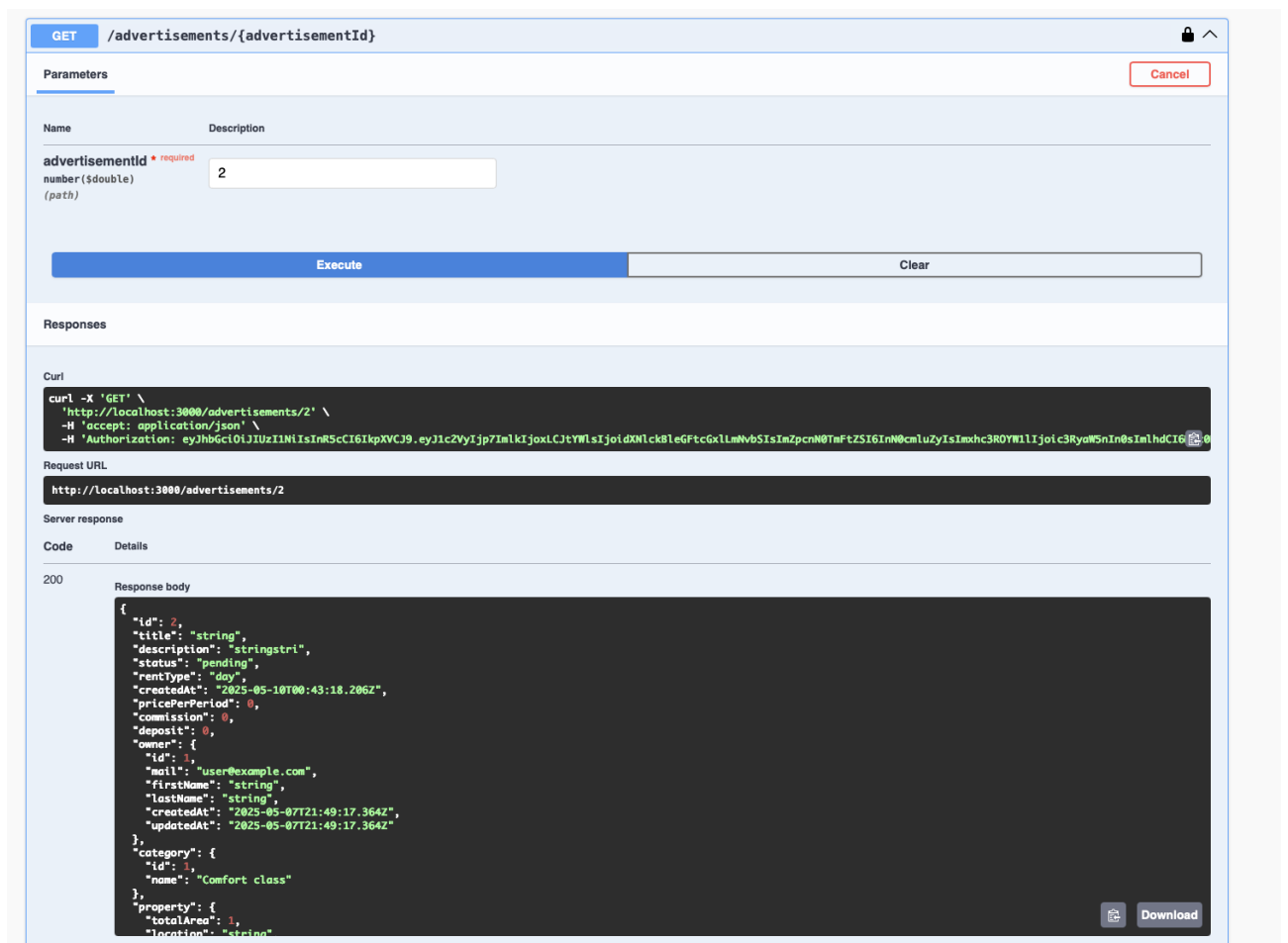


Рисунок 2 – Метод для получения объявления в swagger

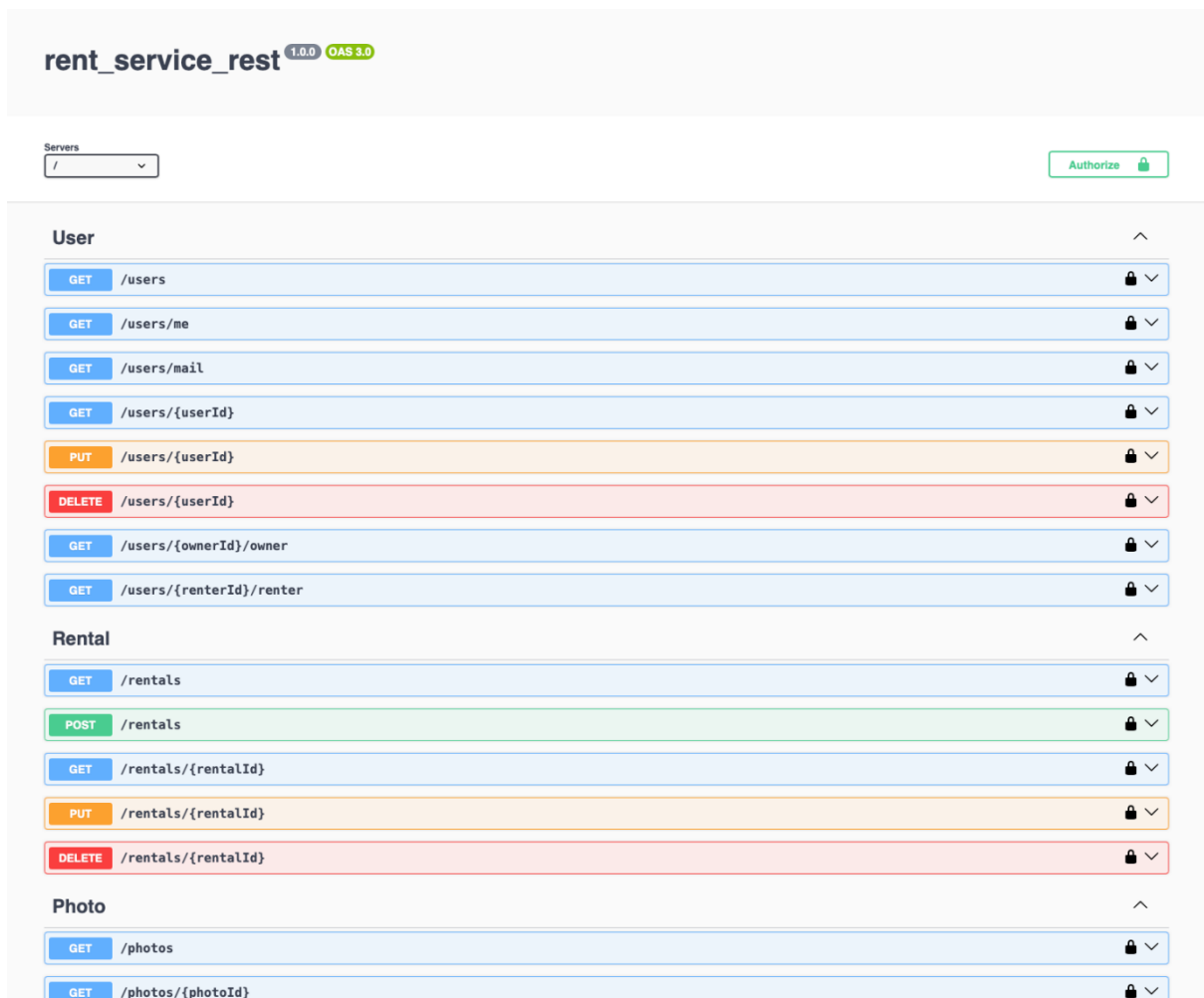


Рисунок 3 – Swagger



Рисунок 4 – Примеры моделей в Swagger

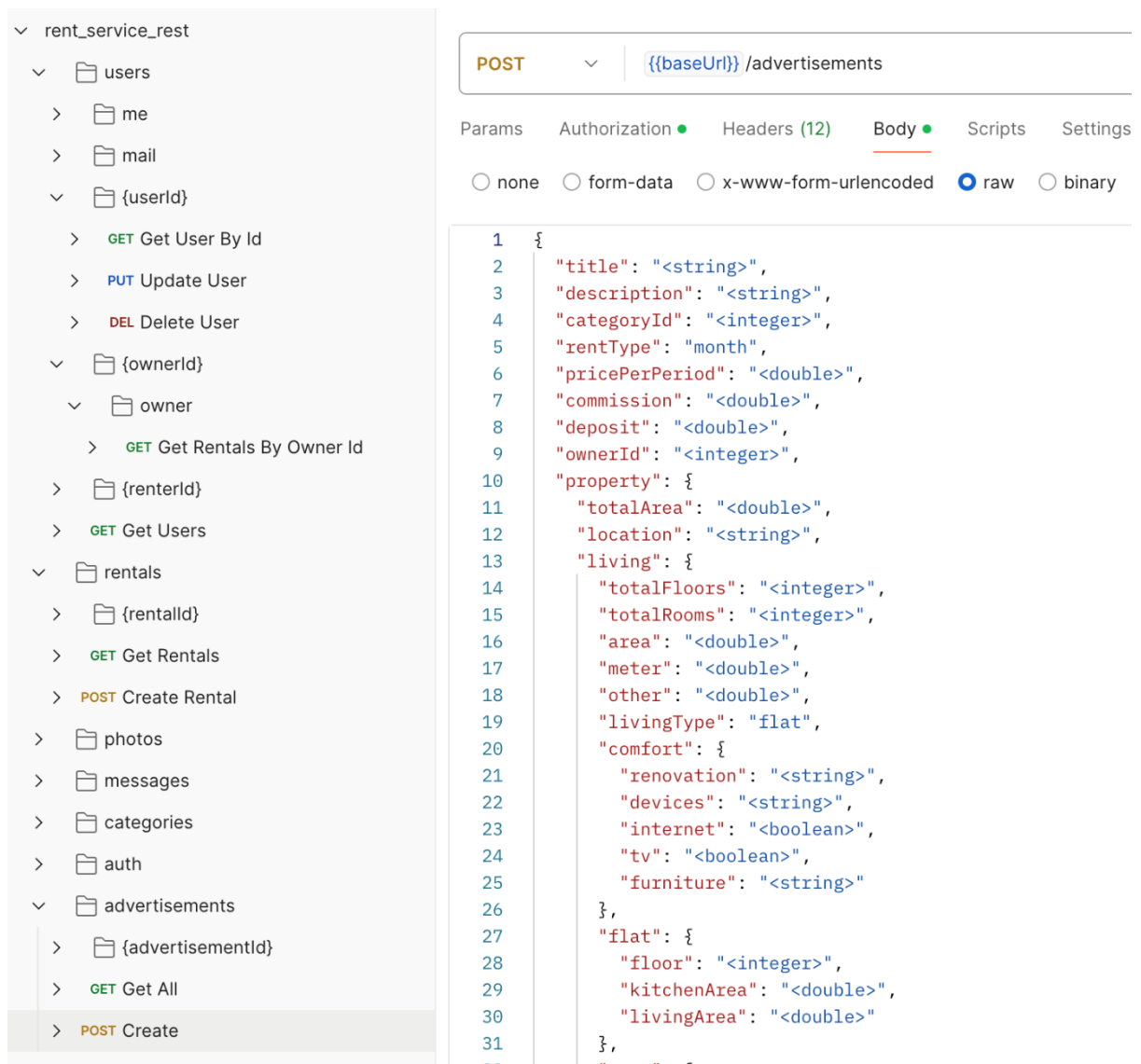


Рисунок 5 – Коллекция в PostMan

## Вывод

В рамках работы реализована автодокументация swagger при помощи фреймворка tsoa, а также создана коллекция запросов в postman.