

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Домашняя работа №3

Выполнила:

Платонова Александра

Группа К3339

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

## Задача

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

## Ход работы

На первом этапе была произведена настройка swagger-jsdoc и swagger-ui-express, добавлены JSDoc-комментарии к роутам, чтобы автоматически генерировать Swagger-документацию.

```
import swaggerJsdoc from 'swagger-jsdoc';
import swaggerUi from 'swagger-ui-express';

const options = {
  definition: {
    openapi: '3.0.0',
    info: {
      title: 'Hostel server',
      version: '1.0.0',
      description: 'API documentation',
    },
  },
  apis: ['./src/routes/*.ts'],
};

const specs = swaggerJsdoc(options);
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(specs));
```

Далее была добавлена конфигурация Swagger.

```
import swaggerJsdoc from 'swagger-jsdoc';
import swaggerUi from 'swagger-ui-express';
import { Express } from 'express';

const options = {
  definition: {
    openapi: '3.0.0',
    info: {
      title: 'Hostel Management API',
      version: '1.0.0',
      description: 'API для управления общежитиями, комнатами, проживающими и платежами',
    },
  },
  servers: [
    { url: 'http://localhost:3000', description: 'Development server' },
  ],
  components: {
    schemas: {
      Address: {
        type: 'object',
        properties: {
          id: { type: 'integer' },
```

```

        city_district: { type: 'string' },
        street: { type: 'string' },
        zip_code: { type: 'string' },
    },
},
Hostel: {
    type: 'object',
    properties: {
        id: { type: 'integer' },
        name: { type: 'string' },
        house_num: { type: 'integer' },
        building: { type: 'integer' },
    },
},
},
},
},
apis: ['./src/controllers/*.ts'],
};

export const setupSwagger = (app: Express) => {
    const specs = swaggerJsdoc(options);
    app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(specs));
};

```

К каждому контроллеру необходимо добавить JSDoc:

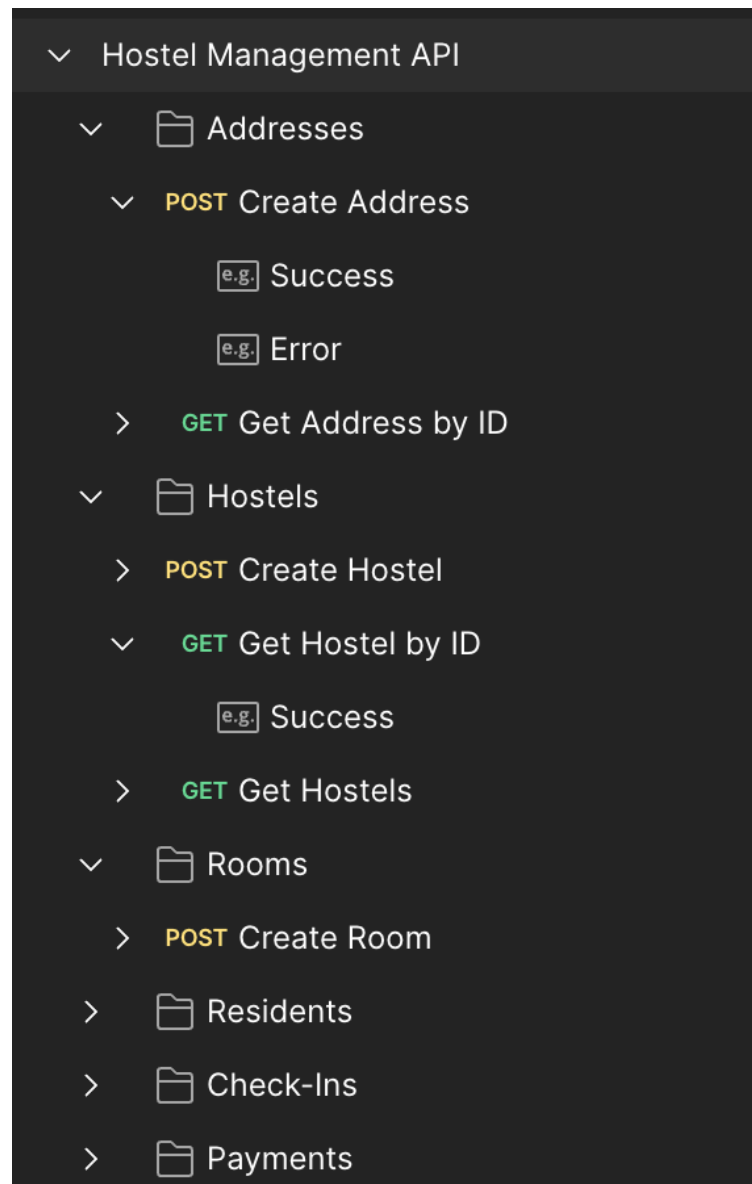
```

/**
 * @swagger
 * tags:
 *   name: Addresses
 *   description: Управление адресами
 */
export class AddressController {
    private repository = new AddressRepository();

    /**
     * @swagger
     * /addresses:
     *   post:
     *     summary: Создать новый адрес
     *     tags: [Addresses]
     *     requestBody:
     *       required: true
     *       content:
     *         application/json:
     *           schema:
     *             $ref: '#/components/schemas/Address'
     *     responses:
     *       201:
     *         description: Адрес успешно создан
     *         content:
     *           application/json:
     *             schema:
     *               $ref: '#/components/schemas/Address'
     *       500:
     *         description: Ошибка сервера
     */
    async create(req: Request, res: Response) {}
}

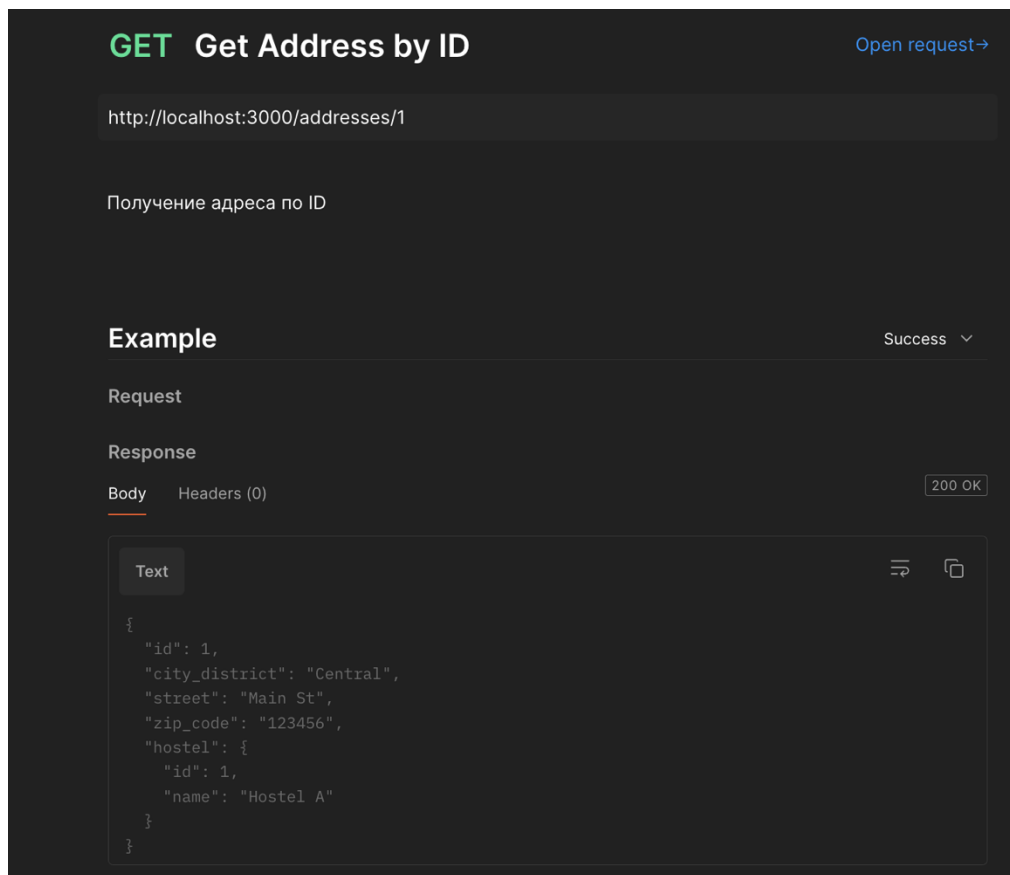
```

На следующем этапе была создана коллекция запросов в Postman (рис. 1), включающая в себя ответы сервера в случае успешного выполнения запроса и в случае серверной ошибки.



*Рисунок 1 - Коллекция Postman*

На основе этой коллекции генерируется документация, пример документированного запроса представлен на рисунке 2.



*Рисунок 2 - Пример документации запроса*

## Вывод

В ходе выполнения домашнего задания была реализована автодокументация средствами Swagger и средствами Postman.