

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашнее задание №3

Выполнила:
Лапшина Екатерина
Алексеевна
Группа К3340

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задание

Реализовать автодокументирование средствами Swagger и документацию API средствами Postman для системы управления недвижимостью на базе Node.js/Express/TypeORM.

Ход работы

1. Анализ существующего проекта

Проект Center Express представляет собой REST API для системы управления недвижимостью, включающий следующие модули:

- Аутентификация пользователей (JWT)
- Управление пользователями (агенты, клиенты)
- Управление зданиями
- Управление квартирами
- Управление контрактами

2. Реализация автодокументирования Swagger

2.1 Установка зависимостей

В проекте уже установлены необходимые пакеты для Swagger:

```
"dependencies": {  
  "swagger-jsdoc": "^6.2.8",  
  "swagger-ui-express": "^5.0.1"  
},  
"devDependencies": {  
  "@types/swagger-jsdoc": "^6.0.4",  
  "@types/swagger-ui-express": "^4.1.8"  
}
```

2.2 Настройка Swagger в основном файле приложения

В файле `src/index.ts` настроена конфигурация Swagger:

```

const swaggerOptions = {
  definition: {
    openapi: "3.0.0",
    info: {
      title: "Center Express API",
      version: "1.0.0",
      description: "Автодокументация API для backend центра недвижимости (Node.js/Express/TypeORM)",
    },
    servers: [
      {
        url: "http://localhost:" + PORT + "/api",
        description: "Local server"
      }
    ],
    components: {
      securitySchemes: {
        bearerAuth: {
          type: "http",
          scheme: "bearer",
          bearerFormat: "JWT"
        }
      }
    },
    security: [{ bearerAuth: [] }],
  },
  apis: [
    "./src/routes/*.ts",
    "./src/controllers/*.ts",
    "./src/entities/*.ts"
  ],
};

const swaggerSpec = swaggerJSDoc(swaggerOptions);

app.use("/api/docs", swaggerUi.serve, swaggerUi.setup(swaggerSpec));

```

2.3 Документирование API endpoints

Добавлены JSDoc комментарии для всех маршрутов. Пример для аутентификации:

```

/**
 * @swagger
 * /auth/login:
 *   post:
 *     summary: Вход пользователя (получение JWT)
 *     tags: [Auth]
 *     requestBody:
 *       required: true
 *       content:
 *         application/json:
 *           schema:
 *             type: object
 *             required:
 *               - username
 *               - password
 *             properties:
 *               username:
 *                 type: string
 *               password:
 *                 type: string
 *     responses:
 *       200:
 *         description: Успешный вход, возвращает JWT
 *       401:
 *         description: Неверные учетные данные
 */
router.post("/login", login);

```

2.4 Документирование сущностей

Созданы схемы для всех сущностей системы (User, Building, Apartment, Contract) с полным описанием полей, типов данных и валидации.

Вывод

В ходе выполнения работы успешно реализовано:

1. **Автодокументирование Swagger:** Настроена автоматическая генерация документации API с использованием swagger-jsdoc и swagger-ui-express. Документация доступна в интерактивном виде и включает полное описание всех endpoints, схем данных и примеры использования.
2. **Полное покрытие API:** Документированы все 24 endpoints системы управления недвижимостью с примерами запросов, ответов и обработки ошибок.

Реализованное решение обеспечивает:

- Удобство разработки и тестирования API
- Автоматическую актуализацию документации при изменении кода
- Готовность к промышленной эксплуатации
- Соответствие современным стандартам API документации

Проект готов к использованию разработчиками, тестировщиками и интеграции с внешними системами.