САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа №3

Выполнил:

Кадникова Екатерина

Группа К3341

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

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

Ход работы

1. Документация средствами Swagger

Настроен Swagger для автоматической генерации документации API (см. Листинг 1). Настроен маршрут для доступа к документации через Swagger UI - http://localhost:3000/api-docs/ (см. Рисунок 1).

Листинг 1 - swagger.ts:

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

const options = {
   definition: {
       openapi: "3.0.0",
       info: {
           title: "Rental API Documentation",
           version: "1.0.0",
            description: "API documentation for Rental Property Management
System",
       },
       servers: [
           {
               url: "http://localhost:3000",
               description: "Development server",
           },
       ],
       components: {
           securitySchemes: {
               bearerAuth: {
                   type: "http",
                   scheme: "bearer",
                   bearerFormat: "JWT",
               },
           },
       },
   },
   apis: ["./src/controllers/*.ts", "./src/models/*.ts"],
};

const specs = swaggerJsdoc(options);
```

```
export const setupSwagger = (app: Express) => {
    app.use("/api-docs", swaggerUi.serve, swaggerUi.setup(specs));

    const router = Router();
    router.get("/api-docs.json", (req, res) => {
        res.setHeader("Content-Type", "application/json");
        res.send(specs);
    });

    app.use(router);
};
```

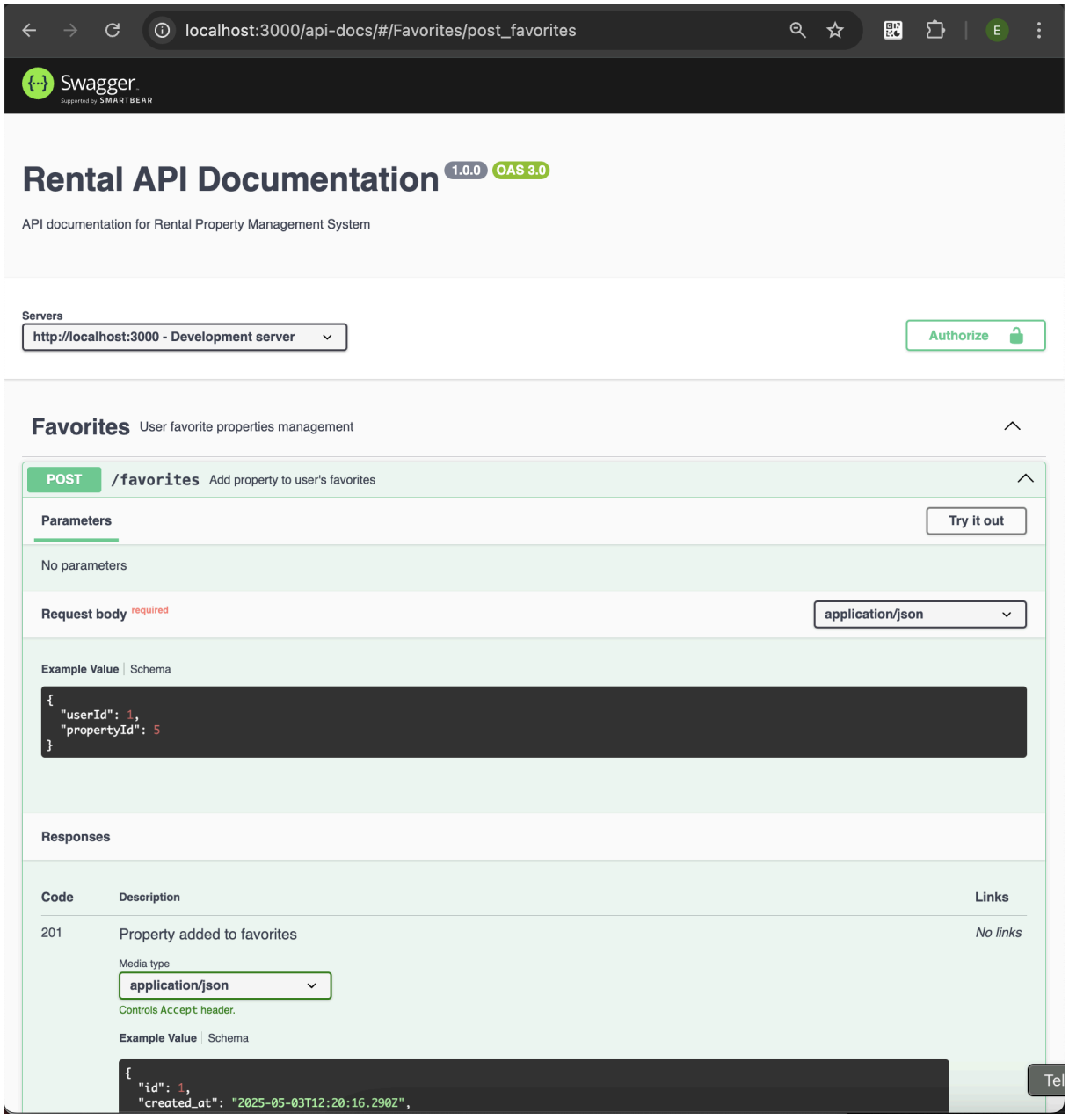

Рисунок 1 - Swagger UI

Реализованы JSDoc комментарии для всех моделей и контроллеров (см. Листинг 2).

Листинг 2 - Пример части задокументированного контроллера:

```
/**
* @swagger
* /users:
*   post:
*     summary: Register a new user
*     tags: [Users]
*     requestBody:
*       required: true
*       content:
*         application/json:
*           schema:
*             $ref: '#/components/schemas/UserCreate'
*     responses:
*       201:
*         description: User created successfully
*         content:
*           application/json:
*             schema:
*               $ref: '#/components/schemas/UserCreateResponse'
*       400:
*         description: Bad request (username or email already in use)
*         content:
*           application/json:
*             schema:
*               $ref: '#/components/schemas/Error'
*       500:
*         description: Internal server error
*         content:
*           application/json:
*             schema:
*               $ref: '#/components/schemas/Error'
*/
export const createUser = async (req: Request, res: Response):
Promise<void> => {
    try {
        const { username, email, password } = req.body;

        const existingUser = await userRepo.findOneBy([{ username }, {
email }]);
        if (existingUser) {
            res.status(400).json({ message: "Username or email already in
use" });
            return;
        }

        const hashedPassword = await bcrypt.hash(password, 10);
```

```
            const user = userRepo.create({ username, email, password_hash:
hashedPassword });
            await userRepo.save(user);

            res.status(201).json({ message: "User created successfully", id:
user.id });
        } catch (err) {
            res.status(500).json({ message: "Error creating user", error: err
});
        }
    };
```

Также обеспечен доступ к сырому Swagger JSON (/api-docs.json).

Все endpoints задокументированы с описанием:

- Параметров запроса;
- Тела запроса;
- Возможных ответов;
- Кодов состояния.

Документация включает примеры запросов и ответов.

2. Документация API в Postman

Экспортирован Swagger JSON (api-docs.json) из документации. Импортирована документация в Postman (см. Рисунок 2). Настроены коллекции запросов для всех endpoints. Добавлены примеры запросов и ответов. Настроена авторизация (JWT токен).

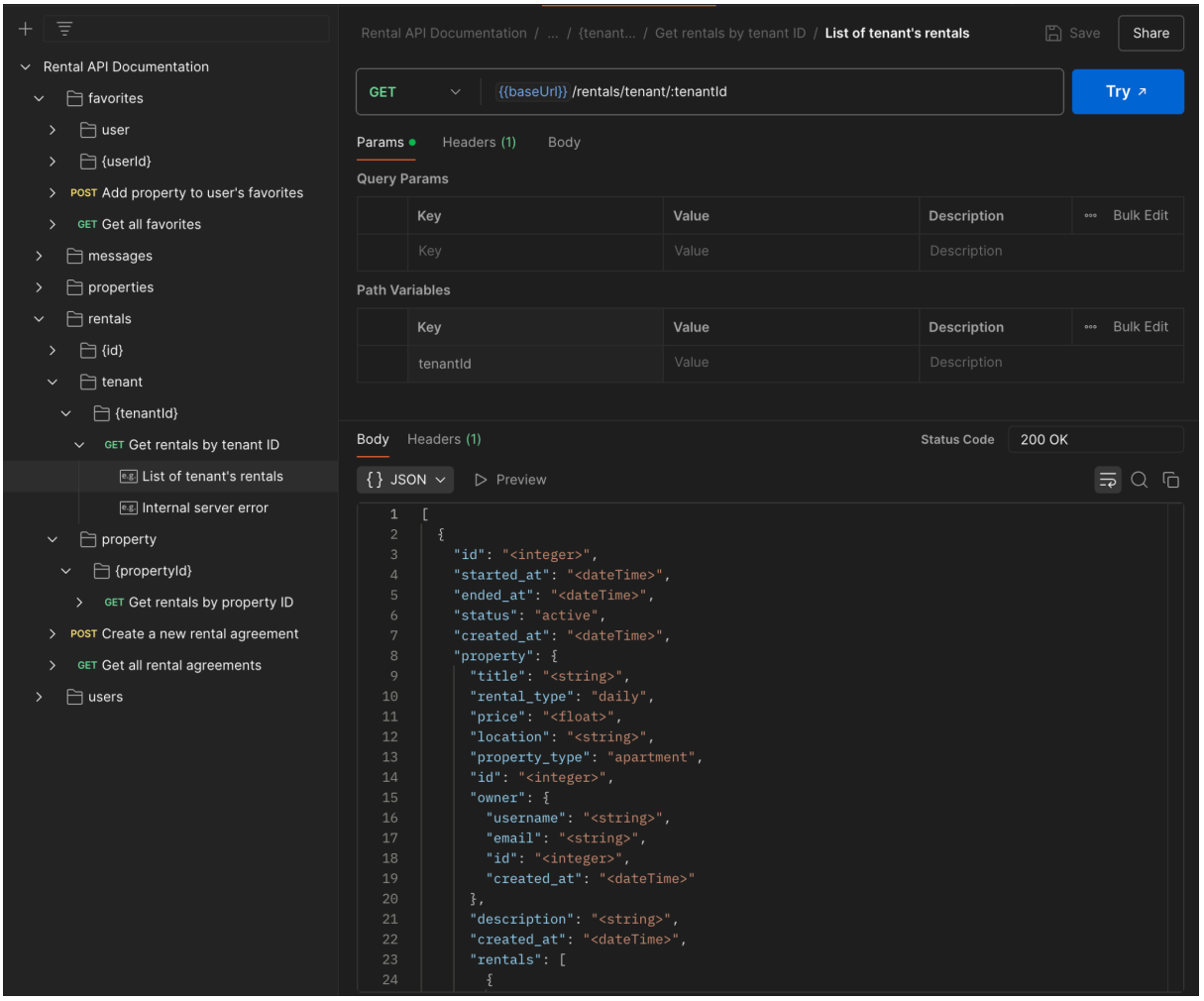

Рисунок 2 - Настроенная коллекция в Postman

Вывод

В рамках работы реализовано автоматическое документирование API с использованием Swagger, включающее все модели и контроллеры системы. Настроен интерактивный интерфейс документации и экспорт в Postman для удобного тестирования. Документация охватывает все эндпоинты с описанием параметров, примеров запросов и возможных ответов, что значительно упрощает взаимодействие с API для разработчиков.