

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа 3

Выполнил:

Сахно Ярослав

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

Установлен пакет Swagger для Laravel API (см. Листинг 1). Настроен маршрут для доступа к документации через Swagger UI - <http://127.0.0.1:8000/api/documentation> (см. Рисунок 1).

Листинг 1 - пакеты composer:

```
"require": {  
    "php": "^8.2",  
    "darkaonline/l5-swagger": "^8.6",  
    "guzzlehttp/guzzle": "^7.2",  
    "laravel/framework": "^10.10",  
    "laravel/sanctum": "^3.2",  
    "laravel/tinker": "^2.8",  
    "league/flysystem-sftp-v3": "^3.15",  
    "treblle/security-headers": "^0.0.3"  
},
```

Homework System API 1.0.0 OAS 3.0

<http://127.0.0.1:8000/docs?api-docs.json>

Документация REST-эндпоинтов Homework System

Filter by tag

Admin Users Управление пользователями через админ-панель

GET **/users** Получить HTML-страницу со списком пользователей

POST **/users** Обработка формы создания пользователя

GET **/users/create** Форма создания пользователя

GET **/users/{id}/edit** Форма редактирования пользователя

Рисунок 1 - Swagger UI

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

Листинг 2 - Пример части задокументированного контроллера:

```
/**
 * @OA\Tag(
 *     name="Admin Users",
 *     description="Управление пользователями через
 админ-панель"
 * )
 */
class UserController extends Controller
{
```

```

/**
 * Список пользователей (HTML-таблица)
 *
 * @OA\Get(
 *     path="/users",
 *     tags={"Admin Users"},
 *     summary="Получить HTML-страницу со списком
пользователей",
 *     @OA\Parameter(
 *         name="type",
 *         in="query",
 *         required=false,
 *         description="Фильтр по user_type_id
(2-преподаватель, 3-студент...)",
 *         @OA\Schema(type="integer")
 *     ),
 *     @OA\Response(
 *         response=200,
 *         description="HTML страница",
 *         content={
 *             @OA\MediaType(
 *                 mediaType="text/html",
 *                 @OA\Schema(type="string",
example="<html>...</html>")
 *             )
 *         }
 *     )
 * )

```

Также обеспечен доступ к сырому Swagger JSON (/api-docs.json). Все endpoints задокументированы с описанием:

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

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.

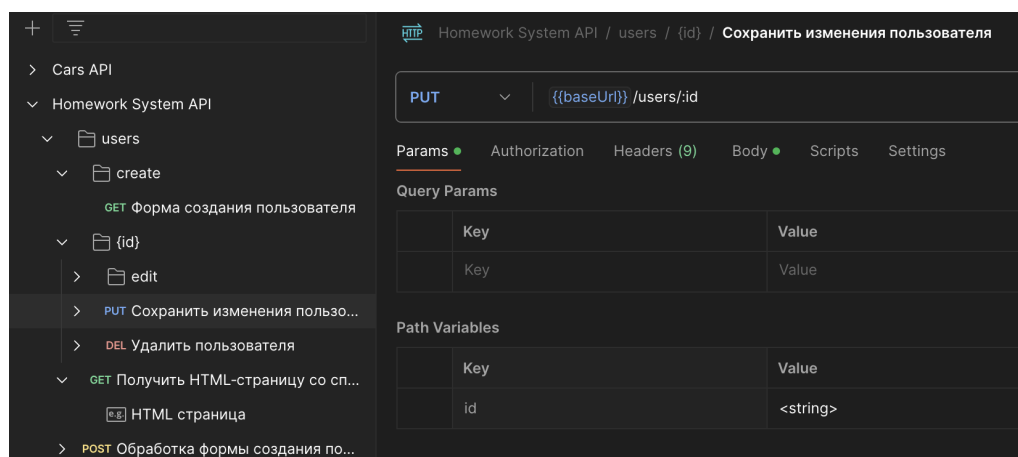


Рисунок 2 - Настроенная коллекция в Postman

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.