

Documentación de Endpoints en Laravel con Swagger/OpenAPI

Esta guía explica cómo documentar endpoints en un proyecto Laravel usando anotaciones OpenAPI (Swagger), qué significa cada parte y qué tan necesario es cada elemento. Incluye ejemplos básicos y avanzados.

¿Qué es Swagger/OpenAPI?

Es un estándar para documentar APIs. Permite describir endpoints, parámetros, respuestas y más, generando documentación interactiva y fácil de entender.

Ejemplo de Anotación Completa

```
/**
 * @OA\Get(
 *     path="/api/promociones", // 1
 *     summary="Obtener promociones", // 2
 *     description="Devuelve una lista de promociones. Si se pasa el parámetro
showAll=true, devuelve todas las promociones, de lo contrario solo las activas.", // 3
 *     tags={"Promociones"}, // 4
 *     @OA\Parameter( // 5
 *         name="showAll",
 *         in="query",
 *         description="Mostrar todas las promociones (true: todas, false u omitido:
solo activas)",
 *         required=false,
 *         @OA\Schema(
 *             type="string",
 *             enum={"true", "false"}
 *         )
 *     ),
 *     @OA\Response( // 6
 *         response=200,
 *         description="Lista de promociones",
 *         @OA\JsonContent(
 *             type="array",
 *             @OA\Items(
 *                 @OA\Property(property="id", type="integer", example=1),
 *                 @OA\Property(property="titulo", type="string", example="Promo
especial"),
 *                 @OA\Property(property="subtitulo", type="string",
example="Subtitulo promo"),
 *                 @OA\Property(property="imagen", type="string",
example="http://localhost:8000/storage/imagen.jpg"),
 *                 @OA\Property(property="estado", type="integer", example=1)
 *             )
 *         )
 *     )
 *)
```

```
* )  
*/
```

Explicación de cada parte

1. **path** : La URL del endpoint (necesario).
2. **summary** : Breve resumen del endpoint (recomendado).
3. **description** : Descripción detallada (opcional pero útil).
4. **tags** : Agrupa endpoints en la interfaz de Swagger (opcional pero recomendable para organización).
5. **@OA\Parameter** : Parámetro que el endpoint recibe (aquí, uno tipo query llamado `showAll`).
Solo si tu endpoint espera parámetros.
6. **@OA\Response** : Describe respuestas del endpoint.
 - `response=200` : Código de respuesta HTTP.
 - `description` : Descripción de la respuesta.
 - **@OA\JsonContent** : Estructura del JSON que devuelve. Siempre debes documentar al menos una respuesta.

¿Es necesario poner todo esto para cada endpoint?

- **NO** necesitas todos los detalles siempre, pero cuantos más detalles, mejor la documentación.
- **Mínimo necesario:**
 - Método (`@OA\Get`, `@OA\Post`, etc.)
 - `path`
 - Al menos una respuesta (`@OA\Response`)
- Parámetros y ejemplos de respuesta **son opcionales** pero muy útiles para quienes consuman la API.

Ejemplo Básico (Mínimo)

```
/**  
 * @OA\Get(  
 *     path="/api/promociones",  
 *     @OA\Response(response=200, description="OK")  
 * )  
 */  
  
public function index(Request $request) {  
    // ...  
}
```

Esto funciona, pero la documentación será muy básica.

Ejemplo Intermedio (Con Parámetro)

```

/**
 * @OA\Get(
 *     path="/api/productos",
 *     summary="Listado de productos",
 *     @OA\Parameter(
 *         name="categoria",
 *         in="query",
 *         description="Filtrar por categoría",
 *         required=false,
 *         @OA\Schema(type="string")
 *     ),
 *     @OA\Response(
 *         response=200,
 *         description="Lista de productos"
 *     )
 * )
 */
public function index(Request $request) {
    // ...
}

```

Ejemplo Completo (Con Parámetros y Respuesta Detallada)

```

/**
 * @OA\Post(
 *     path="/api/usuarios",
 *     summary="Crear usuario",
 *     @OA\RequestBody(
 *         required=true,
 *         @OA\JsonContent(
 *             required={"nombre", "email"},
 *             @OA\Property(property="nombre", type="string", example="Juan"),
 *             @OA\Property(property="email", type="string",
 * example="juan@dominio.com"),
 *             @OA\Property(property="password", type="string", format="password",
 * example="123456")
 *         )
 *     ),
 *     @OA\Response(
 *         response=201,
 *         description="Usuario creado",
 *         @OA\JsonContent(
 *             @OA\Property(property="id", type="integer", example=1),
 *             @OA\Property(property="nombre", type="string", example="Juan"),
 *             @OA\Property(property="email", type="string",
 * example="juan@dominio.com")
 *         )
 *     )
 * )
 */
public function store(Request $request) {

```

```
// ...  
}
```

Tips

- Los **parámetros** solo se documentan si el endpoint los usa.
 - La **respuesta** puede ser simple (solo el código HTTP) o tan detallada como quieras.
 - Usa **tags** para agrupar endpoints relacionados (por ejemplo, "Usuarios", "Productos").
 - Los **ejemplos** ayudan mucho a otros desarrolladores a entender qué esperar de tu API.
-

Recomendación

- Usa descripciones y ejemplos detallados en los endpoints más importantes o públicos.
 - Para endpoints internos, documenta lo básico.
 - Si tu API será consumida por terceros, más detalle = menos dudas.
-

Recursos

- [OpenAPI Specification](#)
 - [L5 Swagger para Laravel](#)
 - [Editor Online de Swagger](#)
-

¿Dudas o quieres ejemplos de otros métodos o casos? ¡Avísame!