

Anexo C Swagger UI de documentación del backend

La documentación de los endpoints REST del backend ha sido generada de forma automática mediante la librería **Springdoc OpenAPI 2**. Esta herramienta proporciona una interfaz visual basada en Swagger UI, que permite consultar y probar los diferentes servicios expuestos por el sistema directamente desde el navegador, sin necesidad de herramientas externas como Postman.

Acceso

Una vez desplegada la aplicación backend en entorno local o remoto, se puede acceder al panel Swagger desde la siguiente URL:

<http://localhost:8080/swagger-ui.html>

Funcionalidades principales

Desde esta interfaz se pueden realizar las siguientes acciones:

- Explorar todos los endpoints disponibles organizados por controlador.
- Consultar el método HTTP (GET, POST, PUT, DELETE) asociado a cada operación.
- Ver los parámetros requeridos en las peticiones, así como los modelos de datos esperados.
- Probar directamente las peticiones REST desde el navegador, introduciendo datos y viendo las respuestas reales del servidor (incluso errores si los hay).
- Autenticarse con un token JWT para acceder a rutas protegidas.

Captura de pantalla

A continuación se muestra una captura de la interfaz Swagger correspondiente a este proyecto:

Swagger UI

OpenAPI definition ([GATSBY](#))

Servers <http://localhost:8080> - Generated server url [Authorize](#)

user-controller

- PUT** /api/users/update Actualizar un usuario
- POST** /api/users/add Crear un nuevo usuario
- GET** /api/users Listar todos los usuarios
- GET** /api/users/{id} Obtener un usuario por ID
- DELETE** /api/users/delete/{id} Eliminar un usuario

role-controller

- PUT** /api/roles/update Actualizar un rol
- POST** /api/roles/add Crear un nuevo rol
- GET** /api/roles Listar todos los roles
- GET** /api/roles/{id} Obtener un rol por ID
- DELETE** /api/roles/delete/{id} Eliminar un rol

inventory-controller

- PUT** /api/inventory/update/{id} Actualizar producto en el inventario
- POST** /api/inventory/import-csv Importar CSV de inventario
- POST** /api/inventory/add Añadir producto al inventario
- GET** /api/inventory/get Listar inventario
- GET** /api/inventory/get/{warehouseId} Listar inventario por almacén
- GET** /api/inventory/generate-csv Generar CSV de inventario
- DELETE** /api/inventory/delete/{id} Eliminar inventario

transaction-controller

- POST** /api/transactions/add Añadir entrada de stock (add)
- GET** /api/transactions/get Listar todas las transacciones

auth-controller

- POST** /api/auth/login Autenticar usuario

warehouse-controller

- GET** /api/warehouses/get Listar almacenes
- GET** /api/warehouses/check/{warehouseId}/{productId} Comprobar si un producto está en un almacén

report-controller

- GET** /api/reports/estadisticas/pdf Descargar reporte de estadísticas en formato PDF

product-controller

- GET** /api/products/get Listar todos los productos
- GET** /api/products/get/{id} Obtener un producto por ID

Inventory-alert-controller

- GET** /api/inventory-alerts/get Listar alertas de inventario

category-controller

- GET** /api/categories/get Obtener todas las categorías

audit-log-controller

- GET** /api/audit-logs Obtener todos los logs de auditoría
- GET** /api/audit-logs/action/{action} Obtener logs por acción

Schemas

- RoleDTO >
- UserDTO >
- WarehouseDTO >
- Role >
- CategoryDTO >
- InventoryDTO >
- ProductDTO >
- TransactionDTO >
- JwtRequest >
- JwtResponse >
- InventoryAlertDTO >
- AuditLogDTO >

Configuración

La configuración de Swagger se encuentra en la clase `SwaggerConfig.java`, ubicada en el paquete `config` del backend. Allí se definen los detalles básicos del proyecto y el esquema de seguridad para permitir autenticación JWT mediante el botón **Authorize**.

```
@Configuration  
  
@SecurityScheme (  
  
    name = "bearerAuth",  
  
    type = SecuritySchemeType.HTTP,  
  
    scheme = "bearer",  
  
    bearerFormat = "JWT",  
  
    in = SecuritySchemeIn.HEADER  
  
)  
  
public class SwaggerConfig {  
  
}
```

Esta configuración permite que Swagger reconozca la necesidad de autenticación para ciertos endpoints y facilite el uso del token correspondiente durante las pruebas.