<dependency>

<groupId>org.springdoc</groupId>

<artifactId>springdoc-openapi-ui</artifactId>

<version>2.0.0</version>

</dependency>

import io.swagger.v3.oas.annotations.Operation;

import io.swagger.v3.oas.annotations.responses.ApiResponse;

import io.swagger.v3.oas.annotations.responses.ApiResponses;

import io.swagger.v3.oas.annotations.tags.Tag;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/books")

@Tag(name = "Books", description = "Operations related to books")

public class BookController {

@Autowired

private BookService bookService;

@GetMapping

@Operation(summary = "Get all books", description = "Retrieve a list of all books")

@ApiResponses(value = {

@ApiResponse(responseCode = "200", description = "Books retrieved successfully"),

@ApiResponse(responseCode = "500", description = "Internal server error")

})

public List<Book> getAllBooks() {

return bookService.getAllBooks();

}

@PostMapping

@Operation(summary = "Add a new book", description = "Create a new book entry")

@ApiResponses(value = {

@ApiResponse(responseCode = "201", description = "Book created successfully"),

@ApiResponse(responseCode = "400", description = "Invalid input data")

})

public Book addBook(@RequestBody Book book) {

return bookService.addBook(book);

}

@GetMapping("/{id}")

@Operation(summary = "Get a book by ID", description = "Retrieve a book by its ID")

@ApiResponses(value = {

@ApiResponse(responseCode = "200", description = "Book retrieved successfully"),

@ApiResponse(responseCode = "404", description = "Book not found")

})

public Book getBookById(@PathVariable Long id) {

return bookService.getBookById(id);

}

}

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springdoc.core.GroupedOpenApi;

import org.springdoc.core.properties.SpringDocProperties;

@Configuration

public class SwaggerConfig {

@Bean

public GroupedOpenApi api() {

return GroupedOpenApi.builder()

.group("v1")

.pathsToMatch("/\*\*")

.build();

}

@Bean

public SpringDocProperties springDocProperties() {

SpringDocProperties properties = new SpringDocProperties();

properties.setApiDocsPath("/api-docs");

return properties;

}

}