Add Swagger Dependency

Maven Dependency

<dependency>

<groupId>org.springdoc</groupId>

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

<version>1.7.0</version>

</dependency>

Document Endpoints

Annotate REST Controllers

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

import io.swagger.v3.oas.annotations.media.Content;

import io.swagger.v3.oas.annotations.media.Schema;

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.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/api/books")

@Tag(name = "Books", description = "API for managing books")

public class BookController {

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

@ApiResponses(value = {

@ApiResponse(responseCode = "200", description = "Found the book",

content = { @Content(mediaType = "application/json",

schema = @Schema(implementation = BookDTO.class)) }),

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

})

@GetMapping("/{id}")

public BookDTO getBookById(@PathVariable Long id) {

// Method implementation

}

@Operation(summary = "Create a new book")

@ApiResponses(value = {

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

content = { @Content(mediaType = "application/json",

schema = @Schema(implementation = BookDTO.class)) }),

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

})

@PostMapping

public BookDTO createBook(@RequestBody BookDTO bookDTO) {

// Method implementation

}

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

@GetMapping

public List<BookDTO> getAllBooks() {

// Method implementation

}

// Additional CRUD operations

}

Annotate Models(DTOs)

import io.swagger.v3.oas.annotations.media.Schema;

@Schema(description = "Data Transfer Object for a Book")

public class BookDTO {

@Schema(description = "Unique identifier of the book", example = "1", required = true)

private Long id;

@Schema(description = "Title of the book", example = "The Catcher in the Rye", required = true)

private String title;

@Schema(description = "Author of the book", example = "J.D. Salinger", required = true)

private String author;

@Schema(description = "Price of the book in USD", example = "15.99", required = true)

private double price;

// Getters and setters

}