

# Swagger

## Table of Contents

1. Introduction .....	1
2. Step 1: Add Dependency .....	1
3. Step 2: Run the Application .....	1
4. Step 3: Access Swagger UI .....	2
5. Step 4: Customize Swagger Info (Optional) .....	2
6. Step 5: Use Annotations to Document Endpoints .....	2
7. Conclusion .....	3
8. Bonus Tips .....	3

## 1. Introduction

Swagger (now called OpenAPI) is a powerful tool for generating interactive documentation for REST APIs. Spring Boot supports Swagger integration through the springdoc-openapi library.

Benefits of using Swagger:

- ¥ Automatically generates API documentation from your code.
- ¥ Provides an interactive UI to test endpoints.
- ¥ Improves API discoverability and developer experience.

## 2. Step 1: Add Dependency

If you're using Maven, add the following dependency in your `pom.xml`:

```
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-ui</artifactId>
  <version>1.7.0</version>
</dependency>
```

## 3. Step 2: Run the Application

No extra configuration is needed for basic setup. Just start your Spring Boot app using:

```
./mvnw spring-boot:run
```

## 4. Step 3: Access Swagger UI

Once the application is running, open your browser and go to:

```
http://localhost:8080/swagger-ui.html
```

You will see an interactive web page listing all your REST endpoints, grouped by controller.

## 5. Step 4: Customize Swagger Info (Optional)

You can customize the API metadata using the `@OpenAPIDefinition` annotation.

```
@OpenAPIDefinition(
    info = @Info(
        title = "Book API",
        version = "1.0",
        description = "REST API for managing books",
        contact = @Contact(name = "Your Name", email = "you@example.com")
    )
)
@SpringBootApplication
public class BookApplication {
    public static void main(String[] args) {
        SpringApplication.run(BookApplication.class, args);
    }
}
```

## 6. Step 5: Use Annotations to Document Endpoints

You can add descriptions and parameter info using annotations like `@Operation` and `@Parameter`.

```
@RestController
@RequestMapping("/api/books")
public class BookController {

    @Operation(summary = "Get all books")
    @GetMapping
    public List<Book> getAllBooks() {
        return bookService.findAll();
    }

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

```
    public Book createBook(@RequestBody Book book) {  
        return bookService.save(book);  
    }  
}
```

## 7. Conclusion

Swagger + Spring Boot makes documenting and testing REST APIs simple and effective. Use it to improve the usability, discoverability, and maintainability of your API.

## 8. Bonus Tips

¥ You can also generate static OpenAPI JSON/YAML files using:

```
curl http://localhost:8080/v3/api-docs
```

¥ For more advanced configurations, explore:

- ! Security schemes
- ! Grouped APIs
- ! Custom themes for Swagger UI