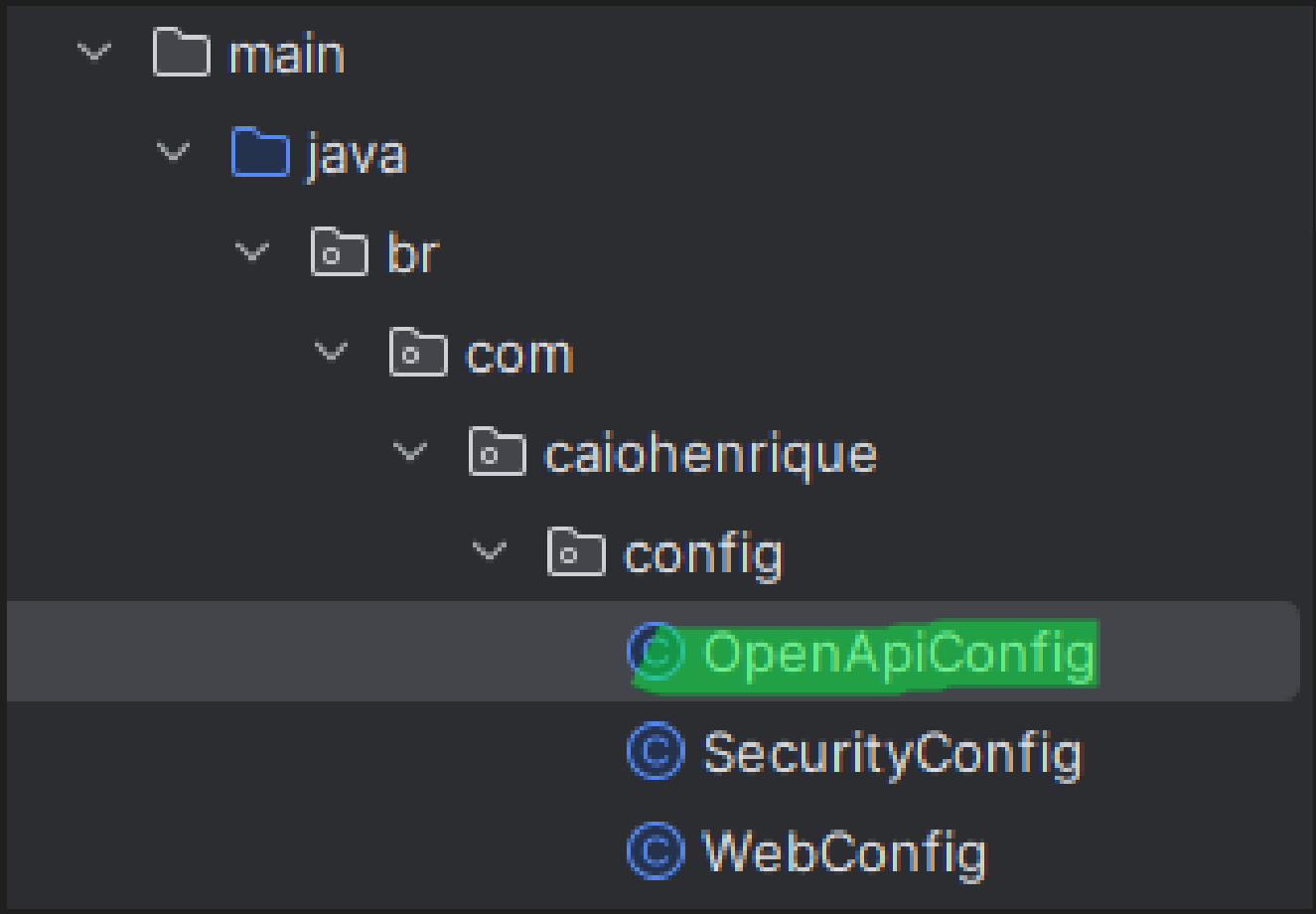
Documenting REST API
Spring Boot
with
SWAGGER

@caiohenriquerm

```xml
<!-- Swagger -->
<dependency>
    <groupId>org.springdoc</groupId>
    <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
    <version>2.2.0</version>
</dependency>
```

To begin with, we will add the openapi dependency for our project, in version 2.2.0.

Don't forget to ensure that Maven has loaded all the necessary dependencies to move forward.



In your Spring application, create a configuration package and then create the OpenApiConfig class.

```java
@Configuration
public class OpenApiConfig {
```

Add the @Configuration annotation to tell Spring that this is a configuration class.

```java
@Bean
OpenAPI customOpenApi() {
```

Create the customOpenApi method with the @Bean annotation, within it you will define all the information about your API, which will appear in the Swagger interface.

Swagger Documentation

```java
@Bean
OpenAPI customOpenApi() {

    return new OpenAPI().info(new Info().
            title("RESTful API by Caio Henrique with Spring Boot.").
            version("v1").
            description("My first contact with API REST and RESTful.").
            termsOfService("https://github.com/caiohrmm/RESTwithSpring").
            license(new License().
                    name("Apache 2.0").
                    url("https://github.com/caiohrmm/RESTwithSpring")));
}
```

Pass all the information about your API into the method parameters, however you see fit.

Configuring Swagger in Controller

After configuring the general Swagger class, we also need to configure all of our endpoints, to clearly define the functions of our API.

```java
@RestController
@RequestMapping("/books/v1")
@Tag(name = "Library", description = "Endpoints for Managing Library")
public class BookController {
    @Autowired
    private BookService service;
```

Add an @Tag annotation, where we will pass the name of what the endpoint manages and its description.

Configurando o Endpoint

```
@GetMapping(value = "/{id}", produces = { APPLICATION_JSON, APPLICATION_XML, APPLICATION_YML})
@Operation(summary = "Find a book", description = "Find a book", tags = {"Library"},
        responses = {
                @ApiResponse(description = "Success", responseCode = "200",
                        content = {
                                @Content(
                                        schema = @Schema(implementation = BookVO.class)
                                )
                        }),
                @ApiResponse(description = "No Content", responseCode = "204", content = @Content),
                @ApiResponse(description = "Bad Request", responseCode = "400", content = @Content),
                @ApiResponse(description = "Unauthorized", responseCode = "401", content = @Content),
                @ApiResponse(description = "Not Found", responseCode = "404", content = @Content),
                @ApiResponse(description = "Internal Server Error", responseCode = "500", content = @Content),
        })
public BookVO findBookById(@PathVariable(value = "id") Long id) { return service.findById(id); }
```

This endpoint searches for a book by Id, I need to show Swagger what types of data this endpoint receives and produces.
I also need to add the @Operation annotation, in which I will define all possible responses from my API to the request and their respective Status Code.

@caiohenriquerm

Configuring the heart of Spring.

After configuring our entire controller, we just need one more adjustment to the heart of Spring (application.properties or application.yml).

application.yml

```yaml
springdoc:
  swagger-ui:
    use-root-path: true
  paths-to-match:
    - /**/v1/**
    - /auth/**
```

application.properties

```properties
springdoc.swagger-ui.use-root-path=true
springdoc.paths-to-match=/**/v1/**, /auth/**
```

Add the routes Swagger needs to document, remembering that ** means "anything".

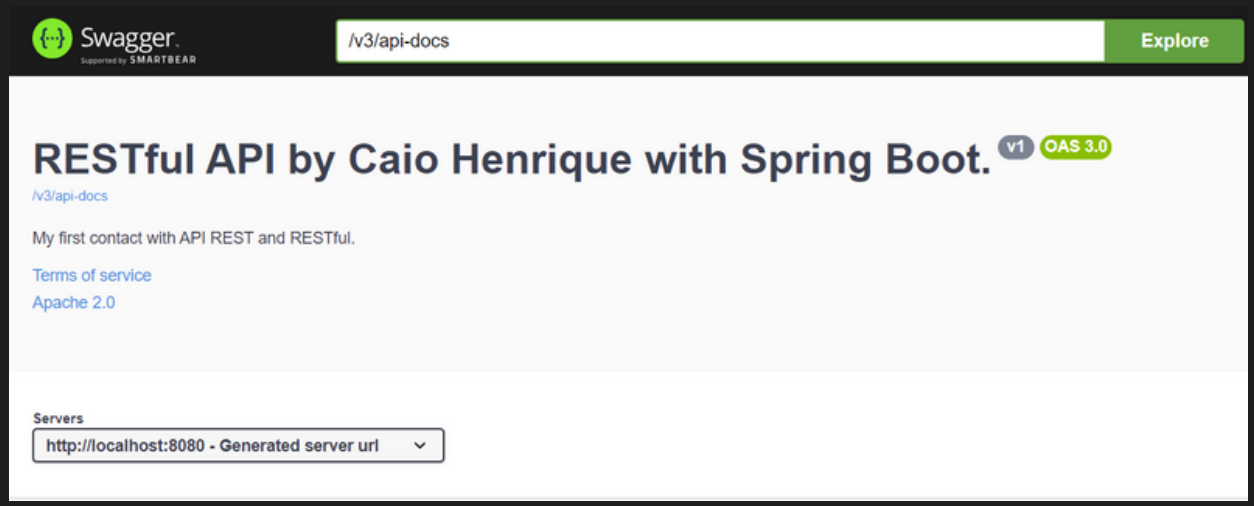
After completing all the steps, start your API and access http://localhost:8080/swagger-ui/index.html to see the Swagger interface.

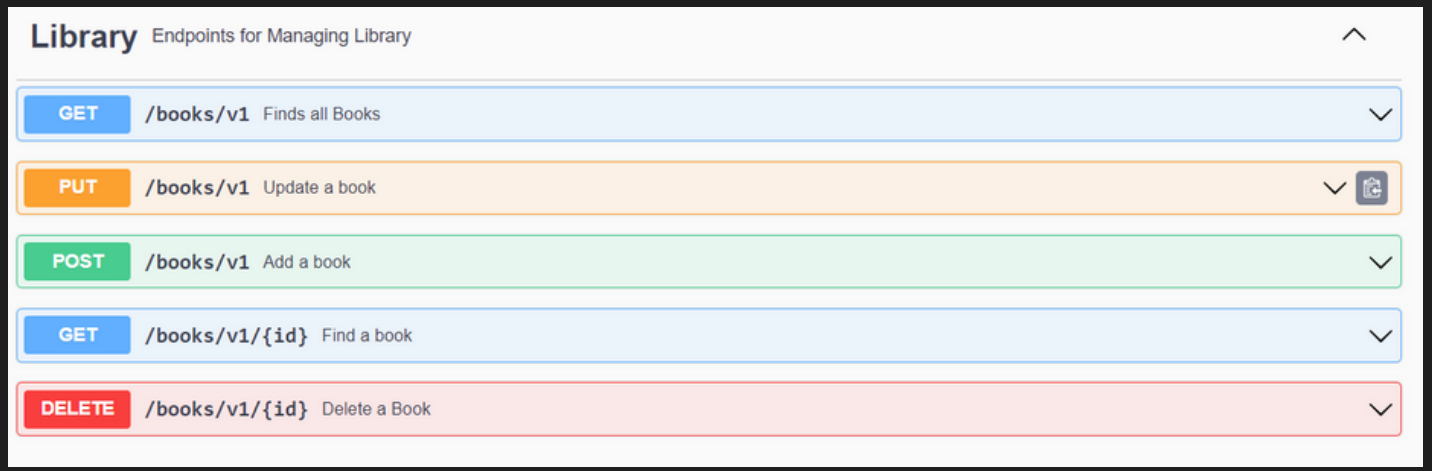
Or

http://localhost:8080/v3/api-docs to access JSON from your API.

@caiohenriquerm

Swagger Interface





@caiohenriquerm

DID YOU LIKE THE CONTENT?

Leave your feedback

To help me improving!

@caiohenriquerm