**Instalación**

Para lograr esto, es requerido solamente dos pasos, agregar las librerías de Swagger y crear un objeto de configuración llamado Docket, veamos cómo sería:

Lo primero es agregar las siguientes librerías en el archivo pom.xml

**Código**

pom.xml

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version>2.9.2</version>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger-ui</artifactId>

<version>2.9.2</version>

</dependency>

La primera (springfox-swagger2) es la más importante, pues es que se encarga de analizar la estructura de nuestro proyecto y crear los metadatos para crear la auto documentación del API. La segunda (springfox-swagger-ui) es que se encarga de tomar los metadatos de la primera y crear una interfaz gráfica amigable con los usuarios.

Configuración

El segundo paso es crear el archivo Docket, mediante el cual le indicamos a Swagger que es lo que debe de tomar en cuanta al momento de crear la documentación. Para esto, debemos de crear una clase que este anotada con @Configuration y @EnableSwagger2 y tenga un método que cree el objeto Docket

SwaggerConfig.java

@Configuration

@EnableSwagger2

public class SwaggerConfig {

@Bean

public Docket apiDocket() {

return new Docket(DocumentationType.SWAGGER\_2)

.select()

.apis(RequestHandlerSelectors.basePackage("com.codmind.

swaggerapi.controllers"))

.paths(PathSelectors.any())

.build()

.apiInfo(getApiInfo())

;

}

private ApiInfo getApiInfo() {

return new ApiInfo(

"Order Service API",

"Order Service API Description",

"1.0",

"http://codmind.com/terms",

new Contact("Codmind", "https://codmind.com", "apis@codmind.com"),

"LICENSE",

"LICENSE URL",

Collections.emptyList()

);

}

}

Dentro del método apiDocket construimos el objeto Docket, en el cual le indicamos que tome todos los servicios que se encuentre en el paquete com.codmind.swaggerapi.controllers y los auto documente por nosotros.

Finalmente, en el método getApiInfo, creamos el objeto ApiInfo, el cual define los datos de la propiedad del API, como su nombre, correo de contacto, licencia, etc.