

Swagger



► <https://swagger.io/>

► dasturlash.uz

How to add

Version - V2 || io.springfox <= 2.X.0

```
<dependency>  
  <groupId>io.springfox</groupId>  
  <artifactId>springfox-swagger-ui</artifactId>  
  <version>2.9.2</version>  
</dependency>
```

```
<dependency>  
  <groupId>io.springfox</groupId>  
  <artifactId>springfox-swagger2</artifactId>  
  <version>2.9.2</version>  
</dependency>
```

<http://localhost:8080/swagger-resources> - get info
<http://localhost:8080/v2/api-docs> - json format
<http://localhost:8080/swagger-ui.html> - ui

How to add

- Version - V3 || io.springfox >= 3.0.0

```
<dependency>
```

```
  <groupId>io.springfox</groupId>
```

```
  <artifactId>springfox-boot-starter</artifactId>
```

```
  <version>3.0.0</version>
```

```
</dependency>
```

<http://localhost:8080/swagger-resources> - get info

<http://localhost:8080/v3/api-docs> - json format

<http://localhost:8080/swagger-ui/index.html> - ui

New Name **OpenAPI 3**

Configuration 1

► How to configure

```
@Configuration
@EnableSwagger2
public class SwaggerConfig {
    @Bean
    public Docket swaggerConfiguration() {

        Set<String> consumes = new HashSet<>();
        consumes.add(MediaType.APPLICATION_JSON_VALUE);

        return new Docket(DocumentationType.SWAGGER_2)
            .consumes(consumes)
            .produces(consumes)
            .select()
            .apis(RequestHandlerSelectors.basePackage("com.company"))
            .build()
            .apiInfo(apiDetails());
    }
}
```

Configuration 2

► Project detail

```
private ApiInfo apiDetails() {  
    return new ApiInfo("My Super Project Name",  
        "Api documentation for project name.",  
        "1.0",  
        "Beast team.",  
        new springfox.documentation.service.Contact("", "", ""),  
        "",  
        "",  
        Collections.emptyList()  
    );  
}
```

Springfox has bug

- ▶ Temper solition:
 - ▶ `spring.mvc.pathmatch.matching-strategy=ant_path_matcher`
- ▶ Or user previous version like 2.6.0

@Api

- ▶ We can add this Annotation to the controller to add basic information regarding the controller.
- ▶ `@Api(tags = "Task")`

@ApiOperation

- This annotation is used to describe the exposed REST API. It describes an operation or typically a HTTP method against a specific path. It takes the following parameters-

Annotation Parameter	Description
value	The value of the annotation is a short description on the API. Since this is displayed in the list of operations in Swagger-UI and the location is limited in size, this should be kept short (preferably shorter than 120 characters)
notes	The notes allows you to give significantly more details about the operations (e.g. you can include request samples and responses here)
nickname	The nickname for this API.

@ApiOperation - example

- ▶ @ApiOperation(value = "getGreeting", notes="get greeting",
nickname = "getGreeting")

@RequestMapping(method = RequestMethod.GET, value = "/api/javainuse")

```
public <Hello> sayHello() {  
    ArrayList<Hello> arrayList= new ArrayList<>();  
    arrayList.add(new Hello());  
    return arrayList;  
}
```

@ApiResponse-

- ▶ This annotation is used to describe the expected responses for the REST API.
- ▶ The @ApiResponse describes a concrete possible response.
- ▶ It cannot be used directly on the method and needs to be included in the array value of @ApiResponses (whether there's one response or more). It takes the following parameters-

Annotation Parameter	Description
ApiResponse	The @ApiResponse describes a concrete possible response

@ApiResponses- Example

```
► @ApiOperation(value = "getGreeting", nickname = "getGreeting")
   @ApiResponses(value = {
       @ApiResponse(code = 500, message = "Server error"),
       @ApiResponse(code = 404, message = "Service not found"),
       @ApiResponse(code = 200, message = "Successful retrieval",
           response = Hello.class, responseContainer = "List") })
   @RequestMapping(method = RequestMethod.GET, value = "/api/javainuse")
   public <Hello> sayHello() {

       ArrayList<Hello> arrayList= new ArrayList<>();
       arrayList.add(new Hello());
       return arrayList;
   }
```

@ApiParam

- This annotation is used to describe the exposed REST API. It takes the following parameters-

Annotation Parameter	Description
value	The value is a short description of the parameter
required	If the parameter is optional or required.
defaultValue	Specify defaultValue of the parameter.

@ApiParam - Example

```
▶ @ApiOperation(value = "getGreeting", nickname = "getGreeting")
   @RequestMapping(method = RequestMethod.GET, value = "/api/javainuse")
   public List<Hello> sayHello(
       @ApiParam(value = "testId", required = true, defaultValue = "111")
       @PathVariable(ID) final int instituionId) {
       return new Hello();
   }
```

@ApiModel

- ▶ The @ApiModel allows you to manipulate the meta data of a model from a simple description or name change to a definition of polymorphism.
- ▶ We have used it to create a response class Hello with default values. It takes the following parameters-
- ▶ This annotation is used in the Model property to add some description to the Swagger output for that model attribute. e.g.

Annotation Parameter	Description
value	The name displayed for the Model class
description	The description of the model class

@ApiModelProperty-

- ▶ The @ApiModelProperty allows controlling Swagger-specific definitions such as allowed values, and additional notes.
- ▶ It also offers additional filtering properties in case you want to hide the property in certain scenarios.
- ▶ We use this parameter for specifying default values to the Response model class Hello. It takes the following parameters-

Annotation Parameter	Description
position	The position of the field in the reponse class during display using swagger.
value	The value of the field when using Swagger. For example the default value of the path varaible will be 111 for class Hello.
required	If the field is optional or required.

@ApiModelProperty and @ApiModelPropertyProperty - example

```
▶ public class Hello {  
  
    private int hellold;  
    private String test;  
  
    @ApiModelProperty(position = 1, required = true, value = "1")  
    public int getHellold() { return hellold; }  
  
    public void setHellold(int hellold) { this.hellold = hellold; }  
  
    @ApiModelProperty(position = 2, required = true, value = "helloTest")  
    public String getTest() { return test; }  
  
    public void setTest(String test) { this.test = test; }  
}
```

Links

- ▶ <https://www.codeusingjava.com/boot/swag1>
- ▶ <https://dzone.com/articles/spring-boot-swagger-3-openapi-3-security-exampleba>