



链滴

Swagger2.X 注解

作者: [jianzh5](#)

原文链接: <https://ld246.com/article/1581834280943>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

使用Swagger2时有几个常用注解需要掌握。

常用注解

作用范围	API	使用位置
协议集描述 类上	@Api	用于controller
协议描述 controller的方法上	@ApiOperation	用在c
非对象参数集 在controller的方法上	@ApiImplicitParams	
非对象参数描述 在@ApiImplicitParams的方法里边	@ApiImplicitParam	
对象参数描述 ApiImplicitParams的方法里边，定义接收的参数形式	@ApiParam	用在
描述返回对象的意义 在返回对象类上	@ApiModelProperty	
对象属性 在参数对象的字段上	@ApiModelProperty	
Response集 在controller的方法上	@ApiResponse	
Response @ApiResponse里边	@ApiResponse	用在
Response	@ResponseHeader	

1. @Api标记

Api 用在类上，说明该类的作用。可以标记一个 Controller 类做为 swagger 文档资源，使用方式：
与Controller注解并列使用。 属性配置：

属性名称	备注	是否弃用
value	url的路径值	
tags	如果设置这个值、value的值会被覆盖	
description eprecated	对api资源的描述	@
basePath Deprecated	基本路径可以不配置	
position Deprecated	如果配置多个Api 想改变显示的顺序位置	
produces	For example, "application/json, application/xml"	
consumes	For example, "application/json, application/xml"	
protocols	Possible values: http, https, ws, wss.	

authorizations

高级特性认证时配置

如:

```
@Api(tags = "product模块")
public class ProductController implements ProductFeign {

}
```

2. @ApiOperation标记

ApiOperation: 用在方法上, 说明方法的作用, 每一个url资源的定义,使用方式:

```
@ApiOperation(
    value = "Find purchase order by ID",
    notes = "",
    response = A.class,
    tags = {""})
```

与Controller中的方法并列使用。

属性配置:

属性名称	备注	是否弃用
value	url的路径值	
tags	如果设置这个值、value的值会被覆盖	
description	对api资源的描述	无
basePath	基本路径可以不配置	无
position Deprecated	如果配置多个Api 想改变显示的顺序位置	
produces	For example, "application/json, application/xml"	
consumes	For example, "application/json, application/xml"	
protocols	Possible values: http, https, ws, wss.	
authorizations	高级特性认证时配置	
hidden	配置为true 将在文档中隐藏	
response	返回的对象	
responseContainer 他无效	这些对象是有效的 "List", "Set" or "Map".,	
responseReference	指定对响应类型的引用。指定的引用可以是地的, 也可以是远程的*将按原样使用, 并覆盖任何指定的response()类。	
responseHeaders	响应旁边提供的可能标题列表。	
httpMethod S" and "PATCH"	"GET", "HEAD", "POST", "PUT", "DELETE", "OPTIO	
code	http的状态码 默认 200	
extensions	扩展属性	

notes

注释说明

response

方法的返回值的类型class

如:

```

@ApiOperation(value = "/createCode", notes = "创建文件", response = ResultObj.class)
@RequestMapping(value = "/createCode")
public ResultObj createCode(
    @RequestParam(value = "tab_name", required = true) String tab_name
    , @RequestParam(value = "type", required = true) Integer type
    , @RequestParam(value = "includeSwagger", required = true) Boolean includeSwagger
    , @RequestParam(value = "moduleName", required = true) String moduleName
    , @RequestParam(value = "packageName", required = true) String packageName) {
    return ResultObj.getDefaultResponse(appTableService.listTable(tab_name, null));
}

```

3. @ApiImplicitParams、@ApiImplicitParam、@ApiParam标记

@ApiImplicitParams

只有value一个属性，用来装多个@ApiImplicitParam。

@ApiImplicitParam

用在@ApiImplicitParams注解中，指定一个请求参数的各个方面

属性名称	备注
paramType	参数放在哪个地方
name	参数名称
value	参数代表的含义
dataType	参数类型
dataTypeClass	参数类型
required	是否必要
defaultValue	参数的默认值

paramType:

header-->请求参数的获取: @RequestHeader(代码中接收注解)
 query-->请求参数的获取: @RequestParam(代码中接收注解)
 path (用于restful接口) --> 请求参数的获取: @PathVariable(代码中接收注解)
 body-->请求参数的获取: @RequestBody(代码中接收注解)
 form (不常用))

如

```

@ApiOperation(value = "/createCode", notes = "创建文件", response = ResultObj.class)
@ApiImplicitParams({
    @ApiImplicitParam(name = "表名", value = "tab_name", paramType = "query", dataType = "String", required = true)
    , @ApiImplicitParam(name = "生成代码类别参数", value = "type", paramType = "query", dataType = "Integer", required = true)
})
@RequestMapping(value = "/createCode")
public ResultObj createCode(
    @RequestParam(value = "tab_name", required = true) String tabName
    , @RequestParam(value = "type", required = true, defaultValue = "-1") Integer type){return null;}

```

@ApiParam

用在接收参数中，指定一个请求参数的信息

属性名称	备注
name	要绑定到的请求参数的名称
value	参数的简单描述
required	是否必要
defaultValue	参数的默认值

@ApiParam请求属性,使用方式：与Controller中的方法并列使用。

属性配置：

属性名称	备注
name	属性名称
value	属性值
defaultValue	默认属性值
allowableValues	可以不配置
required	是否属性必填
access	不过多描述
allowMultiple	默认为false
hidden	隐藏该属性
example	举例子

如

```

public ResponseEntity<Order> getOrderById(
    @ApiParam(value = "ID of pet that needs to be fetched", allowableValues = "range[1,5]", required = true)
    @PathVariable("orderId") String orderId)

```

4. @ApiModelProperty、@ApiModelProperty

@ApiModelProperty

描述一个Model的信息（这种一般用在post创建的时候，使用@RequestBody这样的场景，请求参无法使用 @ApiModelProperty注解进行描述的时候；

属性名称	备注
value	名称
description	描述
parent	父类 (class)
discriminator	官方描述：支持模型继承和多态性 这是用作鉴别器的字段名。基于这个领域，可以断言需要使用哪种子类型。
subTypes	子类型 { (class) }
reference	官方描述：指定对相应类型定义的引
, 覆盖指定的任何其他元数据	

@ApiModelProperty

描述一个model的属性。

```
@Table(name = "app_table", schema = "", catalog = "")
@ApiModelProperty(value = "物理表信息表", description = "当前数据库表信息")
public class AppTable implements Serializable {

    private static final long serialVersionUID = -1L;

    @Id
    @Column(name = "tab_name", unique = true)
    @ApiModelProperty(value = "表名主键")
```

5. @ApiResponses、@ApiResponse

@ApiResponses：响应集配置，使用方式：

```
@ApiResponses({ @ApiResponse(code = 400, message = "Invalid Order") })
```

与Controller中的方法并列使用。属性配置：

属性名称	备注
value	多个ApiResponse配置

如：

```
@RequestMapping(value = "/order", method = POST)
```

```

@ApiOperation(value = "Place an order for a pet", response = Order.class)
@ApiResponses({ @ApiResponse(code = 400, message = "Invalid Order") })
public ResponseEntity<String> placeOrder(
    @ApiParam(value = "order placed for purchasing the pet", required = true) Order order) {
    storeData.add(order);
    return ok("");
}

```

@ApiResponse: 响应配置, 使用方式:

```
@ApiResponse(code = 400, message = "Invalid user supplied")
```

与Controller中的方法并列使用。 属性配置:

属性名称	备注
code	http的状态码
message	描述
response	默认响应类 Void
reference	参考ApiOperation中配置
responseHeaders 置说明	参考 ResponseHeader 属性
responseContainer	参考ApiOperation中配置

在SpringMvc中的配置如下:

```

@RequestMapping(value = "/order", method = POST)
@ApiOperation(value = "Place an order for a pet", response = Order.class)
@ApiResponses({ @ApiResponse(code = 400, message = "Invalid Order") })
public ResponseEntity<String> placeOrder(
    @ApiParam(value = "order placed for purchasing the pet", required = true) Order order) {
    storeData.add(order);
    return ok("");
}

```

6. ResponseHeader

响应头设置, 使用方法

```
@ResponseHeader(name="head1",description="response head conf")
```

与Controller中的方法并列使用。 属性配置:

属性名称	备注
name	响应头名称
description	头描述

response
responseContainer

默认响应类 Void
参考ApiOperation中配置