## **swagger2 注解整体说明**

@Api：用在请求的类上，表示对类的说明

tags="说明该类的作用，可以在UI界面上看到的注解"

value="该参数没什么意义，在UI界面上也看到，所以不需要配置"

@ApiOperation：用在请求的方法上，说明方法的用途、作用

value="说明方法的用途、作用"

notes="方法的备注说明"

@ApiImplicitParams：用在请求的方法上，表示一组参数说明

@ApiImplicitParam：用在@ApiImplicitParams注解中，指定一个请求参数的各个方面

name：参数名

value：参数的汉字说明、解释

required：参数是否必须传

paramType：参数放在哪个地方

· header --> 请求参数的获取：@RequestHeader

· query --> 请求参数的获取：@RequestParam

· path（用于restful接口）--> 请求参数的获取：@PathVariable

· body（不常用）

· form（不常用）

dataType：参数类型，默认String，其它值dataType="Integer"

defaultValue：参数的默认值

@ApiResponses：用在请求的方法上，表示一组响应

@ApiResponse：用在@ApiResponses中，一般用于表达一个错误的响应信息

code：数字，例如400

message：信息，例如"请求参数没填好"

response：抛出异常的类

@ApiModel：用于响应类上，表示一个返回响应数据的信息

（这种一般用在post创建的时候，使用@RequestBody这样的场景，

请求参数无法使用@ApiImplicitParam注解进行描述的时候）

@ApiModelProperty：用在属性上，描述响应类的属性

@ApiIgnore()用于类或者方法上，可以不被swagger显示在页面上   
比较简单, 这里不做举例

## **1、@Api：用在请求的类上，说明该类的作用**

@Api：用在请求的类上，说明该类的作用

tags="说明该类的作用"

value="该参数没什么意义，所以不需要配置"

示例：

@Api(tags="APP用户注册Controller")

## **2、@ApiOperation：用在请求的方法上，说明方法的作用**

@ApiOperation："用在请求的方法上，说明方法的作用"

value="说明方法的作用"

notes="方法的备注说明"

示例：

@ApiOperation(value="用户注册",notes="手机号、密码都是必输项，年龄随边填，但必须是数字")1

## **3、@ApiImplicitParams：用在请求的方法上，包含一组参数说明**

@ApiImplicitParams：用在请求的方法上，包含一组参数说明

@ApiImplicitParam：用在 @ApiImplicitParams 注解中，指定一个请求参数的配置信息

name：参数名

value：参数的汉字说明、解释

required：参数是否必须传

paramType：参数放在哪个地方

· header --> 请求参数的获取：@RequestHeader

· query --> 请求参数的获取：@RequestParam

· path（用于restful接口）--> 请求参数的获取：@PathVariable

· body（不常用）

· form（不常用）

dataType：参数类型，默认String，其它值dataType="Integer"

defaultValue：参数的默认值

示列：

@ApiImplicitParams({

@ApiImplicitParam(name="mobile",value="手机号",required=true,paramType="form"),

@ApiImplicitParam(name="password",value="密码",required=true,paramType="form"),

@ApiImplicitParam(name="age",value="年龄",required=true,paramType="form",dataType="Integer")

})

## **4、@ApiResponses：用于请求的方法上，表示一组响应**

@ApiResponses：用于请求的方法上，表示一组响应

@ApiResponse：用在@ApiResponses中，一般用于表达一个错误的响应信息

code：数字，例如400

message：信息，例如"请求参数没填好"

response：抛出异常的类

示例：

@ApiOperation(value = "select1请求",notes = "多个参数，多种的查询参数类型")@ApiResponses({

@ApiResponse(code=400,message="请求参数没填好"),

@ApiResponse(code=404,message="请求路径没有或页面跳转路径不对")

})

## **5、@ApiModel：用于响应类上，表示一个返回响应数据的信息**

@ApiModel：用于响应类上，表示一个返回响应数据的信息

（这种一般用在post创建的时候，使用@RequestBody这样的场景，

请求参数无法使用@ApiImplicitParam注解进行描述的时候）

@ApiModelProperty：用在属性上，描述响应类的属性

示例:

import io.swagger.annotations.ApiModel;import io.swagger.annotations.ApiModelProperty;

import java.io.Serializable;

@ApiModel(description= "返回响应数据")public class RestMessage implements Serializable{

@ApiModelProperty(value = "是否成功")

private boolean success=true;

@ApiModelProperty(value = "返回对象")

private Object data;

@ApiModelProperty(value = "错误编号")

private Integer errCode;

@ApiModelProperty(value = "错误信息")

private String message;

/\* getter/setter \*/

}

[http://localhost:5680/zxmall/swagger-ui.html](http://localhost:5680/zxmall/swagger-ui.html" \t "http://blog.csdn.net/xiaojin21cen/article/details/_blank)