# swagger2jspwiki工具使用文档

## 前言

此工具所要实现目的，通过在类中添加swagger的注释（如下），获取swagger json生成接口文档。

请注意文档中红色字体。

**import** com.jfinal.aop.Clear;  
**import** com.jfinal.core.Controller;  
**import** io.swagger.annotations.\*;  
**import** io.swagger.util.Json;  
**import** java.util.HashMap;  
**import** java.util.Map;  
  
*/\*\*  
 \* Created by LGZ on 2016/7/27.  
 \*/*  
@Api(value = **"/swaggerTest"**, description = **"gets some data from a servlet"**)  
**public class** SwaggerController **extends** Controller {  
  
 @Clear  
 **public void** index() {  
 render(**"/swagger/index.html"**);  
 }  
  
 @io.swagger.annotations.ApiOperation(httpMethod = **"POST"**, value = **"用户登录接口"**, response = Json.**class**, nickname = **"作者昵称"**)  
 @ApiResponses({@ApiResponse(code = 200, message = **"Invalid input"**, response =Json.**class，reference = "{'code':'登录结果','msg':'登录结果描述'}"**)  
 })  
 @ApiImplicitParams({  
 @ApiImplicitParam(name = **"username"**, value = **"User's name"**, required = **true**, dataType = **"string"**, paramType =**"query"**),  
 @Clear  
 **public void** testLogin() {

String name = getPara("username");  
 Map<String, Object> result = new HashMap<String, Object>();

result.put("code", "001");

result.put("msg ", "登录成功");

renderJson(result);  
 }  
 }

## swaggerUi注解使用规范

### 1.1接口所在类的注解

在接口所在类上使用

**Api用法**

| **可用字段** | **必填** | **参数类型** | **描述** |
| --- | --- | --- | --- |
| value | 否 | [string] | 请求地址(路由配置中的地址)，如果不填，会直接走接口地址，如果接口地址是在根目录下则没问题，如果是子目录，则会请求不到 |
| description | 否 | [string] | 描述 |
| tags | 否 | [string] | 配置接口API名称 |

value为finalConfig中的路由配置路径或相对于项目basePath的相对路径；

description为描述，可忽略不填；

tags为当前类中接口所在的标签或模块比如，“用户操作模块（注册/登陆/注销等）”或者“SDK管理模块（新增SDK/修改SDK/获取最新SDK等）”等等。

注:tags内容请至少填写一个，例如tags = "SDK管理相关接口"，多个的话将以第一个为准。如下面示例将会以“用户管理”为准。

**示例：**

@Api(value = **"/swaggerTest"**, description = **"gets some data from a servlet"**, tags = "SDK管理相关接口")

### 1.2接口注解

在接口上使用

**ApiOperation用法**

| **可用字段** | **必填** | **参数类型** | **描述** |
| --- | --- | --- | --- |
| httpMethod | 否 | string | 请求方式，支持的请求类型 "GET", "HEAD", "POST", "PUT", "DELETE", "OPTIONS" and "PATCH". |
| value | 是 | string | 接口名称 |
| response | 否 | Objece.class | 响应类型 |
| nickname | 否 | string | 接口作者（请填写自己姓名或昵称） |

ApiResponse用法

| **可用字段** | **必填** | **参数类型** | **描述** |
| --- | --- | --- | --- |
| code | 是 | int | http协议响应状态码 |
| message | 是 | string | 描述 |
| response | 否 | Objece.class | 响应类型 |
| reference | 否 | string | 参考 |

注意：如果想要告诉阅读文档的人此接口响应的结果代表什么，请加上reference来说明此接口的响应示例，例如reference = "{'code':'登陆结果代码','msg':'登陆结果描述'}"，如果响应的结果是自定义的Object，请参考[1.3自定义的响应类注解](#_1.3自定义的响应类注解)

**示例：**

@ApiOperation(httpMethod = **"GET"**, value = **"用户登录接口"**, response = Json.**class**)  
@ApiResponses({  
 @ApiResponse(code = 200, message = **"登陆成功"**, response = Json.**class**, reference = **"{'code':'登陆结果代码','msg':'登陆结果描述'}"**)  
})  
@ApiImplicitParams({  
 @ApiImplicitParam(name = **"id"**, value = **"User ID"**, required = **false**, dataType = **"integer"**, paramType =**"query"**),  
 @ApiImplicitParam(name = **"name"**, value = **"User's name"**, required = **false**, dataType = **"string"**, paramType = **"query"**),  
 @ApiImplicitParam(name = **"dateOfBirth"**, value = **"User's date of birth, in dd-MM-yyyy format"**, required = **false**, dataType = **"java.util.Date"**, paramType = **"query"**)

})

### 1.3自定义的响应类注解

#### 1.3.1类注解

在自定义的响应类的上方使用

**ApiModel用法（可忽略）**

| **可用字段** | **必填** | **参数类型** | **描述** |
| --- | --- | --- | --- |
| value | 否 | [string] | 类名 |
| description | 否 | [string] | 描述 |

**示例：**

@ApiModel(value=**"用户类"**, description=**"用户的相关参数"**)

#### 1.3.2属性注解

在自定义的响应类的属性上使用

**ApiModelProperty用法**

| **可用字段** | **必填** | **参数类型** | **描述** |
| --- | --- | --- | --- |
| value | 否 | [string] | 属性名 |
| allowableValues | 否 | [string] | 限定值，这个只只允许填什么内容 |

**示例：**

@ApiModelProperty(value = **"用户名"**, allowableValues = **"字母,数字,下划线"**)

## 2.工具配置说明

*#\*\*\*\*\*\*\*\*\*\*\*\*\*\*swagger配置\*\*\*\*\*\*\*\*\*\*\*  
#json来源  
#SWAGGER\_JSON\_URL = http://test.wosdk.cn/wosdkbusiness/swagger-api/swagger.json***SWAGGER\_JSON\_URL** = **http://192.168.1.152:8080/api/swagger.json***#\*\*\*\*\*\*\*\*\*\*\*\*\*jspWiki配置\*\*\*\*\*\*\*\*\*\*\*\***#登录地址***LOGIN\_BASE\_URL** = **http://192.168.1.152:8085/JSPWiki/Login.jsp?redirect=Main***#编辑地址***EDIT\_PAGE\_BASE\_URL** = **http://192.168.1.152:8085/JSPWiki/Edit.jsp***#获取页面地址***GET\_PAGE\_BASE\_URL** = **http://192.168.1.152:8085/JSPWiki/Wiki.jsp***#删除页面地址***DELETE\_PAGE\_BASE\_URL** = **http://192.168.1.152:8085/JSPWiki/Delete.jsp***#每个接口的头路径，因为swaggerJson中获取到的接口路径为相对路径***JIE\_KOE\_BASE\_URL** = **http://test.wosdk.cn/wosdkbusiness***#jspwiki登录账号***LOGIN\_USER\_NAME**=**username***#jspwiki登录密码***LOGIN\_PASSWORD**=**password***#需要保留原页面内容的页面,保留的页面将会追加新内容，而不覆盖旧内容，其他页面旧内容将会丢失  
#LeftMenu为左侧导航栏  
#多个页面用“|”分隔， 例如LeftMenu|swagger1.1***NEED\_RETAIN\_PAGE**=**LeftMenu**