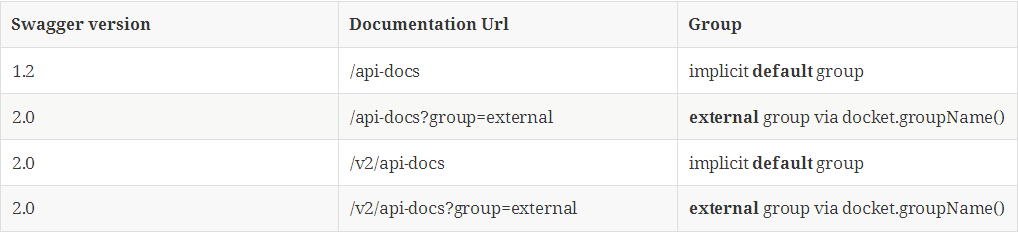
默认的swagger服务描述在下列urls中生成



在index.html中对其进行配置

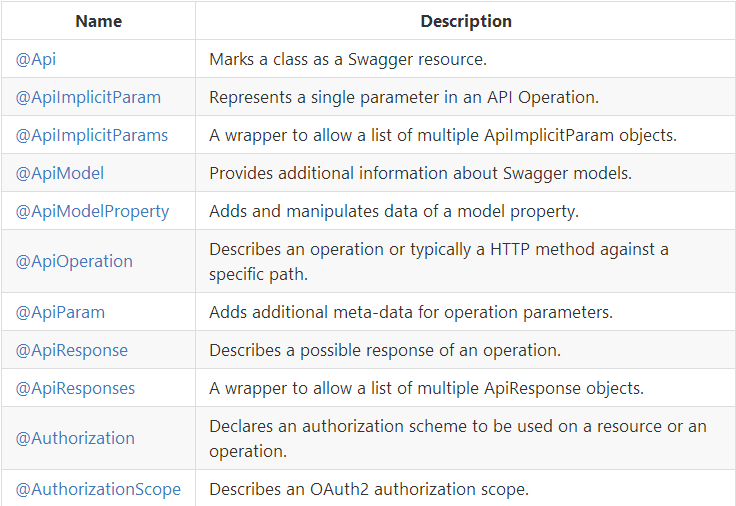
注解分为三类：

声明资源的注解

声明操作的注解

声明API模型的注解

常用的注解



# @Api

Swagger2.0中resources被tags替代；

用于为资源下定义的所有操作提供定义。

注解规范

@Api(value = "/sample/users", description = "gets some data from a servlet", consumes="application/json, application/xml")

public class SampleServlet extends HttpServlet {}

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Value | 资源路径（存在tags则此参数失效） |
| Authorization[] | Authorization | 包含value、scope两个属性，此api资源所需的授权列表和作用范围，可重写 |
| String | Basepath | 由swagger自动导出，可重写。 |
| String | Consumes | Api资源输入类型，如“application/json，application/xml” |
| String | description | 接口介绍 |
| Boolean | Hidden | 是否隐藏接口 |
| Int | Position | 可选的api资源在资源列表中的显示排序 |
| String | Produces | Api资源输出类型，如“application/json，application/xml” |
| String | Tags | 用于Api文档管理的标签列表 |

# @ApiOperation

在api资源中声明一个操作；

操作=path+HTTP Method

注解规范：

@ApiOperation(httpMethod = "GET",

value = "Resource to get a user",

response = SampleData.class,

nickname="getUser")

public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {...}

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Value | 操作说明，可见内容在120字符以内 |
| Authorization[] | Authorization | 包含value、scope两个属性，此api资源所需的授权列表和作用范围，可重写 |
| Boolean | Hidden | 是否隐藏接口 |
| String | Consumes | Api资源输入类型，如“application/json，application/xml” |
| String | HttpMethod | "GET", "HEAD", "POST", "PUT", "DELETE", "OPTIONS" "PATCH" |
| Int | Position | 可选的api资源在资源列表中的显示排序 |
| String | Nickname | 由第三方工具使用，唯一地标识此操作 |
| String | Notes | 操作的详细描述 |
| String | Produces | Api资源输出类型，如“application/json，application/xml” |
| Class<?> | response | 操作的响应类型 |
| String | responseContainer | 标注响应类型是否为一系列的值，"List", "Array", "Set" |
| String | Tags | 用于Api文档管理的标签列表 |
| Int | Code | http响应的状态码，必须为Http定义的状态码中之一 |
| String | ResponseReference | 为响应类型指定一个本地或远程引用用于重写任意指定响应类 |
| responseHeader[] | responseHeaders | 在响应旁提供的可能的头列表 |
| String | Protocols | 指定协议，http, https, ws, wss |

问题：code参数设置后未生效

# @ApiResponses，@ApiResponse

使用HTTP状态码返回错误信息或其他成功的消息；一般操作的返回类型在@ApiOperation中定义，其余的返回码在此注解下定义

注解规范：

@ApiResponses(value = {

@ApiResponse(code = 400, message = "Invalid ID supplied",

responseHeaders = @ResponseHeader(name = "X-Rack-Cache", description = "Explains whether or not a cache was used", response = Boolean.class)),

@ApiResponse(code = 404, message = "Pet not found") })

public Response getPetById(...) {...}

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| Int | Code | 必须参数，状态码 |
| String | Message | 必须参数，信息内容 |
| Class<?> | response | 可选的响应类来描述消息的有效内容 |
| String | Reference | 为响应类型指定一个本地或远程引用用于重写任意指定响应类 |
| String | ResponseContainer | 声明一个容器来装载响应类， "List", "Set" ,"Map" |
| responseHeader[] | responseHeaders | 在响应旁提供的可能的头列表 |

# @Authorization,@AuthorizationScope

这个注解只能在@Api和@ApiOperation中使用。

一旦你已经声明并配置了你在你的应用系统中支持的授权方案，你可以使用这些注解来说明资源或特定操作所需的授权方案。

注解规范：

@ApiOperation(value = "Add a new pet to the store",

authorizations = {

@Authorization(

value="petoauth",

scopes = {

@AuthorizationScope(

scope = "add:pet",

description = "allows adding of pets")

}

)

}

)

public Response addPet(...) {...}

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Value | 必须参数，声明的认证方案 |
| AuthorizationScope[] | scopes | 认证范围 |
| String | Description | 必须参数，范围描述 |
| String | Scope | 必须参数，认证范围具体值 |

# @ApiParam

只能在JAX-RS框架项目中使用

(@PathParam, @QueryParam, @HeaderParam, @FormParam and in JAX-RS 2, @BeanParam)

扫描到上述注解时，@ApiParam注解可用来添加更多细节。

注解规范

@Path("/{username}")

@ApiOperation(value = "Updated user",

notes = "This can only be done by the logged in user.")

public Response updateUser(

@ApiParam(value = "name that need to be updated", required = true) @PathParam("username") String username,

@ApiParam(value = "Updated user object", required = true) User user) {...}

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Access | 允许从api文档里过滤参数 |
| String | Allowablevalues | 限制参数可以接收的值（枚举、范围、最大最小值） |
| Boolean | Allowmultiple | 指定参数是否可以接受多个逗号分隔的值 |
| String | defaultValue | 描述参数的默认值 |
| String | Name | 参数名 |
| Boolean | Required | 指定参数是否必须 |
| String | Value | 参数详细描述 |
| String | Example | 非体类型参数的单例 |
| Boolean | Hidden | 参数是否在参数列表中隐藏 |
| Example | Examples | 参数的样例 |

# @ApiImplicitParam，@ApiImplicitParams

手动描述各种参数，需求情景：

1. 使用servlets不使用JAX-RS注解
2. 由于参数存在重名参数，需要隐藏
3. 描述一个过滤器使用的参数或优先到达JAX-RS实现类中的资源

注解规范：

@ApiImplicitParams({

@ApiImplicitParam(name = "name", value = "User's name", required = true, dataType = "string", paramType = "query"),

@ApiImplicitParam(name = "email", value = "User's email", required = false, dataType = "string", paramType = "query"),

@ApiImplicitParam(name = "id", value = "User ID", required = true, dataType = "long", paramType = "query")

})

public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {...}

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Access | 允许从api文档里过滤参数 |
| String | Allowablevalues | 限制参数可以接收的值（枚举、范围、最大最小值） |
| Boolean | Allowmultiple | 指定参数是否可以接受多个逗号分隔的值 |
| String | defaultValue | 描述参数的默认值 |
| String | Name | 参数名 |
| Boolean | Required | 指定参数是否必须 |
| String | Value | 参数详细描述 |
| String | Datatype | 参数的数据类型，可以是一个类或原生类型 |
| String | ParamType | path, query, body, header,form |
| Example | Examples | 参数的样例 |
| String | Example | 非体类型参数的单例 |

# @ResponseHeader

在@ApiResponse或@ApiOperation中添加，用于描述响应头，包含响应头的名称、描述和类型

注解规范

@ApiResponses(value = {

@ApiResponse(code = 400, message = "Invalid ID supplied",

responseHeaders = @ResponseHeader(name = "X-Rack-Cache", description = "Explains whether or not a cache was used", response = Boolean.class)),

@ApiResponse(code = 404, message = "Pet not found") })

public Response getPetById(...) {...}

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Discription | 对响应头的长文本描述 |
| String | Name | 响应头的名称 |
| Class<?> | Response | 响应头的数据类型 |
| String | responseContainer | 定义容器用于装载响应头，”list”,”set” |

# @ApiModel

允许你通过将一个简单的描述或名称改为一个多态性的定义来操作模型的元数据

实现多态性和继承性

注解规范

@ApiModel(value="SuperModel", discriminator = "foo", subTypes = {SubModel.class})public class SuperModel {...}

@ApiModel(value="SubModel")public class SubModel {...}

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Discription | 为类提供一个更长的描述 |
| String | Discrimination | 支持多态和继承，是一个作为鉴别器的字段名；此字段让应该被使用的子类型生效 |
| Class<?> | parent | 指定父类模型 |
| Class<?>[] | subTypes | 继承模型的子类型列表 |
| String | Value | 为模型提供一个替代名 |
| String | Refrence | 为相应的类型定义指定一个引用，重写所有其他的元数据定义 |

# @ApiModelProperty

允许管理swagger特征定义，如：allowed values,additional notes,additional filter properties

注解规范

@ApiModelProperty(value = "pet status in the store", allowableValues = "available,pending,sold")

public String getStatus() {

return status;

}

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Access | 允许从api文档里过滤参数 |
| String | allowableValues | 限制参数可以接收的值（枚举、范围、最大最小值） |
| String | dataType | 参数的数据类型，可以是一个类或原生类型 |
| Boolean | Hidden | 允许在swagger模型定义中隐藏一个模型属性 |
| Int | Position | 模型属性排序 |
| Boolean | Required | 定义参数是否必须 |
| String | Value | 属性的详细描述 |
| String | Example | 属性样例值 |
| String | Notes |  |
| String | Name | 重写属性名 |
| String | Refrence | 为相应的类型定义指定一个引用，重写所有其他的元数据定义 |
| Boolean | readOnly | 允许将一个模型属性设置为只读 |

# @SwaggerDefinition

提供swagger对象的许多（不是全部）顶层属性，允许你为自动生成的定义进行设置。此注解可以添加在任何在swagger自定义过程中扫描到的类上。另外，它不必在JAX-RS API类上，也可以在一个标记/定义接口上。

注解规范：

@SwaggerDefinition(

info = @Info(

description = "Gets the weather",

version = "V12.0.12",

title = "The Weather API",

termsOfService = "http://theweatherapi.io/terms.html",

contact = @Contact(

name = "Rain Moore",

email = "rain.moore@theweatherapi.io",

url = "http://theweatherapi.io"

),

license = @License(

name = "Apache 2.0",

url = "http://www.apache.org/licenses/LICENSE-2.0"

)

),

consumes = {"application/json", "application/xml"},

produces = {"application/json", "application/xml"},

schemes = {SwaggerDefinition.Scheme.HTTP, SwaggerDefinition.Scheme.HTTPS},

tags = {

@Tag(name = "Private", description = "Tag used to denote operations as private")

},

externalDocs = @ExternalDocs(value = "Meteorology", url = "http://theweatherapi.io/meteorology.html")

)public interface TheWeatherApiConfig {

}

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Basepath | 在生成的swagger定义中指定basePath |
| String[] | Consumes | swagger定义的全局cusume |
| ExternalDocs | ExternalDocs | Swagger定义的外部文档的引用 |
| String | Host | 指定swagger定义中的主机名 |
| Info | Info | 为swagger定义生成元数据 |
| String[] | Produces | swagger定义的全局Produces |
| swaggerDefinition.Scheme[] | Schemes | 接口传输协议：DEFAULT、HTTP、HTTPS、WS、WSS |
| Tag[] | tags | 全局标签用于标记个人api和apiOperation |

存在多个此类注解时，最新的一个注解会将前面的同名注解覆盖

## @info

添加swagger定义中的元数据信息

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Title | 必须参数，Api标题 |
| String | Version | 必须参数，Api版本 |
| Contact | Contact | Api联系信息 |
| String | Description | Api描述信息 |
| Extension[] | Extensions | Api所有扩展信息列表;{properties，name} |
| License | License | Api的license信息 |
| String | termsOfService | Api服务条款 |

### @Contact

Api联系信息

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Name | 必须参数，联系信息名称 |
| String | Email | 联系邮箱 |
| String | Url | 与此联系信息关联的url |

### @License

Swagger定义info部分中有效的许可元数据

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Name | 许可名称 |
| String | Url | 许可的url |

### @Extension

在@ApiOperation、@Info、@Tag中都能使用，用于添加扩展信息

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| ExtensionProperty[] | Properties | 必须参数，扩展属性 |
| String | Name | 这些扩展属性的总名称 |

#### @ExtensionProperty

扩展信息属性

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 说明 |
| String | Name | 属性名 |
| String | Value | 属性值 |