搜索需要查询的内容

登录

接口文档神器Swagger (上篇)

勿忘初心 2018-09-04 09:54

作者: 李哲

接口文档管理一直是一个让人头疼的问题,伴随着各种接口文档管理平台涌现,如阿里开源的rap,ShowDoc, sosoapi, 等等 (网上能找到很多这种管理平台,包括我们自己做的idoc)。这些平台都是一个共同特点,创建文档,编辑,保存文档,一些功能 强大的还有mock,统计接口信息等功能,所以这些平台更像一个接口文档的存储管理系统,可以方便人们查看、编辑文档。然而 接口一般都是经常变化(添加、删除参数),这就需要接口编写者及时更新文档,否则文档将失去意义,但是频繁去更新文档也会 花费开发不少时间。Swagger的出现在某种程度上解决了这些问题,Swagger提供了一套完整的接口文档解决方案,其功能非常强 大,下面将从Swagger简单介绍和使用、Swagger-springmvc原理解析、Swagger其他功能等方面介绍。

一、Swagger介绍和使用

1、什么是swagger

Swagger 是一个规范和完整的框架,用于生成、描述、调用和可视化RESTful风格的 Web 服务。总体目标是使客户端和文件系统 作为服务器以同样的速度来更新。文件的方法,参数和模型紧密集成到服务器端的代码,允许API来始终保持同步。Swagger让部 署管理和使用功能强大的API变得非常简单。官方网站:http://swagger.io/。

Swagger采用OpenAPI规范, OpenAPI规范这类API定义语言能够帮助你更简单、快速的表述API, 尤其是在API的设计阶段作用特 别突出。一旦编写完成, API文档可以作为:

- · 需求和系统特性描述的根据
- · 前后台查询、讨论、自测的基础
- · 部分或者全部代码自动生成的根据
- · 其他重要的作用,比如开放平台开发者的手册

2、如何编写API文档

- (1) 定义YAML文件,然后可以生成各种语言的代码框架,对于后台程序员来说,较少人会愿意写出一堆YAML格式。
- (2) 定义JSON格式文件,按照swagger文档书写规范编写文档,和YAML一样只是两种不同格式。
- (3) 通过swagger的各种语言的插件,可以通过配置及少量代码,生成接口文档及测试界面。通过yaml或json书写的是静态文 档,需要书写文档中需要的内容,详细写法可参考:https://www.gitbook.com/book/huangwenchao/swagger/details,完成后可以 通过可视化页面显示接口文档。但要完成整个项目的接口文档书写也非常耗时,如果是后台开发,可以通过简单配置实现文档的自 动生成。
- 3、Springmvc项目中使用swagger
- 1) 添加maven依赖

```
<dependency>
          <groupId>com. mangofactory</groupId>
          \langle artifactId \rangle swagger-springmvc \langle /artifactId \rangle
          <version>1.0.2
       </dependency>
```

2) 定义一个swagger配置类,添加如下注解,具体内容可参考网上demo

```
@Configuration
@EnableWebMvc
@EnableSwagger
@ComponentScan(basePackages="com.quality.web.deploy")//添加这个注释,会自动扫描该类中的每一个方法自动生成api文档
public class SwaggerConfig {
     private SpringSwaggerConfig springSwaggerConfig;
3) 在spring配置文件中将写的config类添加一个bean
   <bean class="com.quality.core.SwaggerConfig" />
```

4) 在controller中接口处添加swagger注解和相应参数

```
// 初始化发布管理页面需求
例 のまたな事を実施制度を

使RequestMapping("initDeployRequirement")

優ResponseBody

@ApiOperation(value = "初始化支布详情",httpMethod="GET")
         iOperation(value = symm.oc.wrm )
iOperation(value = symm.oc.wrm )
iIc MapcString, Object> getDeployRequirement(
    @ApiParam(value="%EIXd",name="projectId",required=true)
    @RequestParam("projectId") String projectId) {
    MapcString, Object> retMap = new HashMapcString, Object>();
    ListListListLiarReqModel> deployReqList = jiraRequirementServiceImpl.getToDeployReqModelList(projectId);
    if (deployReqList != null && deployReqList.size() > 0) {
        retMap.put("retCode", 200);
        retMap.put("deployReqList", deployReqList);
    } else {
                          retMap.put("retCode", 400);
              return retMap;
```

5) Swagger UI配置

从https://github.com/swagger-api/swagger-ui 获取3.0版本以下, 2.0版本以上。其所有的dist目录下东西放到需要集成的项目里, 本文放入src/main/webapp/WEB -INF/docs/ 目录下。

修改swagger/index.html文件,默认是从连接http://petstore.swagger.io/v2/ swagger.json获取 API 的 JSON,这里需要将url内 容修改为http://{ip}:{port}/{project Name}/api-docs的形式, {}中的内容根据具体情况填写。比如本文的url值为: http://localhost/quality/docs/api-docs。所有工作完成后,在浏览器中输入上述url地址可得到如下页面(注:该页面是swagger官 网示例)。



最新博客

用户奖励体系有哪些反作弊的机制? 手机app有了短信验证码还有没必要有图片 验证码?

【译】AI 让科技公司变得更强大吗 prometheus比zabbix好在哪点?

【盘点2018】社交领域投融资报告来了!

介绍 Scratch 3.0: 扩展编码创造力

【译】如何写出一份优秀的软件设计文档 微服务端点之间的通信 (1) — 编制和编排

微服务架构简介(6)—Spring

【译文】我已经忘了怎么去阅读

【◆】最新资源下载

话社娱乐直播加电商实践-王铃声

直播商业化之路-王昊

网易云新直播内容变现平台-刘勇

实时互动·智能安全——网易娱乐云解决方 案-周森

网易《悠梦》看AR前沿技术--杨鹏

视频的"直播plus"模式-陈太锋

网易易盾助力游戏构建安全护城河--卓辉

网易轻舟,构建敏捷高效的软件生产力体

臧铖-浙商银行"区块链+供应链"双链生产力 志同道合 共绘云图_英特尔 胡凯

■! 编辑推荐



-网打尽! 2018网络 安全事件最全的盘点



Facebook内部报告: 争取青少年用户的鸡贼 小技巧



企业项目开发--maven 父子模块 (1)



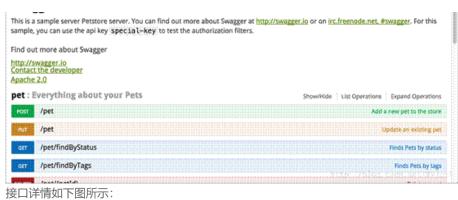
【译文】东京的外国工 程师



HBase最佳实践



手滑把库给删了, 跑路 前应该做的事。。。



初始化发布详情 /deploy/initDeployRequirement Implementation Notes getDeployRequirement Response Class (Status 200) object is not defined! Response Content Type "/" -Parameters projectId (required) Response Messages HTTP Status Code 204 No Content 403 Forbidden default Object is not a primitive Try it out!

可以根据请求参数访问接口,如果网络通畅将返回接口的response结果,在该页面可以看到接口的基本详情,而且如果后台接口发 生变化,该页面中的信息也会随着变化,这样接口的内容就是实时变化的,相对于静态的文档每次改动都需要开发更新文档的方 式,该方式无疑是非常好的解决方案。

如果过程中发现没有达到预期的结果,请检查swagger ui位置是否正确;spring中是否添加了SwaggerConfig的bean。

4、Springboot项目中使用swagger2

相对于springmvc中添加swagger, springboot中更加简单。Springboot中许多配置是默认的,不用太多配置就能完成swagger的接 入。具体流程如下:

1) 添加maven依赖

```
<!-- swagger框架 -->
      <dependency>
          <groupId>io.springfox/groupId>
          \label{localization} $$ \artifactId> springfox-swagger2 </artifactId> $$
          <version>2.2.2
      </dependency>
      <dependency>
          <groupId>io.springfox/groupId>
          \langle artifactId \rangle springfox-swagger-ui \langle /artifactId \rangle
          <version>2.2.2
      </dependency>
```

2) 创建swagger2配置类,该类和springmvc中的不太一样,具体可参考swagger官网。配置中通过@Configuration注解,让 Spring来加载该类配置。再通过@EnableSwagger2注解来启用Swagger2。

```
@Configuration
@EnableSwagger2
public class Swagger2Config {
    public Docket createRestApi() {
         return new Docket(DocumentationType.SWAGGER_2)
                  .apiInfo(apiInfo())
```

3) 和springmvc一样,在controller接口中编写swagger相关的注解来标识接口信息。如下所示,通过@ApiOperation注解来给API 增加说明、通过@ApilmplicitParams、@ApilmplicitParam注解来给参数增加说明。

```
@ApiOperation(value="更新用户详细信息", notes="根据url的id来指定更新对象,并根据传过来的user信息来更新用户详细信息")
@ApiImplicitParams({
          @ApiImplicitParam(name = "id", value = "用户ID", required = true, dataType = "Long"),
@ApiImplicitParam(name = "user", value = "用户详细实体user", required = true, dataType = "User")
@RequestMapping(value="/{id}", method=RequestMethod.PUT)
public String putUser(@PathVariable Long id, @RequestBody UserModel user) {
     return "success";
```

4) 完成上述代码添加上,启动Spring Boot程序,访问:http://localhost:8080/swagger-ui.html。就能看到前文所展示的RESTful API的页面。我们可以再打开具体的API请求,以POST类型的/users请求为例,可找到上述代码中我们配置的Notes信息以及参数 user的描述信息,如下图所示。

登录

注册



博客

专题

问答

资源下载

通过简单的配置改动,spring就能结合swagger,通过简单的方式自动生成接口文档,而且以可视化页面的形式展现,非常灵活方 便。

下面对swagger在spring中使用的一些常用注解进行说明:

直播

- Api, Apilgnore
- ApiModel
- ApiModelProperty
- ApiOperation
- ApiParam、ApiImplicitParam、ApiImplicitParams
- ApiResponse
- ApiResponses
- ResponseHeader
- 1) Api注解用于类上,说明该类的作用。可以标记一个Controller类做为swagger 文档资源。与Controller注解并列使用。

属性如下:

属性名称	备注
value	uri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
basePath	基本路径可以不配置
position	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

2) ApiOperation注解,用在方法上,说明方法的作用,与Controller中的方法并列使用。

属性名称	备注	
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	
basePath	基本路径可以不配置	
position	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	这些对象是有效的 "List", "Set" or "Map"., 其他无效	
httpMethod	"GET", "HEAD", "POST", "PUT", "DELETE", "OPTIONS" and "PATCH	
code	http的状态码 默认 200	
extensions	扩展属性	

3) ApiParam注解用在@ApiImplicitParams的方法里边,属性配置:

登录

搜索需要查询的内容

直播 博客 专题 问答 资源下载

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

4) ApiResponse注解用于响应配置,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	参考 ResponseHeader 属性配置说明
responseContainer	参考ApiOperation中配置

- 5) ApiResponses注解中包含ApiResponse, 用于描述一组ApiResponse值。
- 6) ResponseHeader注解用于设置响应头,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	默认响应类 Void
responseContainer	参考ApiOperation中配置

- 7) ApilmplicitParams注解用于方法上包含一组参数说明。
- 8) ApilmplicitParam注解用于描述请求参数,根据不同的paramType类型,请求的参数来源不同。属性及取值如下:

	取值	属性
查询参数类		paramType
以地址的形式提交数	path	
直接跟參数完成自动缺射戰	query	
以流的形式提交(仅支持POS	body	
參数在request headers 里边提	header	
以form表单的形式提交仅支持POS	form	
参数的数据类型只作为标志说明,并没有实际验证		dataType
	Long	
	String	
接收参数		name
接收参数的意义描述		value
参数是否必		required
951	true	
非必	false	
默认		defaultValue

- 9) ApiModel注解用于表示对类进行说明,描述一个Model的信息,表示参数用实体类接收。
- 10) ApiModelProperty注解用于方法、字段,表示对model属性的说明或者数据操作更改,配合ApiModel一起使用。
- 11) Apilgnore注解用于类或方法上,表示不需要swagger处理。

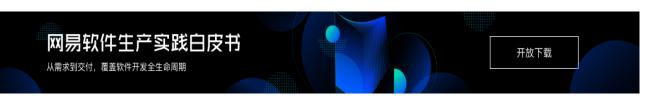
Swagger提供的注解还有其他一些,此处不做解释(而且高版本中的注解可能不太一样),可以通过官方文档或源码查阅,基本常 用的注解就是这些,详细的使用方法可以网上查看。如果熟悉了这些注解就可以很快的表示后台代码接口的信息,然后通过页面的 形式展示出来。但是swagger是如何结合spring通过简单的配置、添加注解的方式就将接口的信息展现出来。下面将看看swagger 为什么这么神奇。

相关阅读:接口文档神器Swagger(下篇) 网易云大礼包: https://www.163yun.com/gift 本文来自网易实践者社区, 经作者李哲授权发布



<上一篇 Mybatis Generator的使用及源码浅析(下...

下一篇接口文档神器Swagger(下篇)>





资源下载

Q

语音交互如何测试与评估?

8月第4周业务风控关注 |国家网信办:坚决将"三俗"内容从网络短视频中清理出去原创:易盾网易易盾 今

专题 问答

接口文档神器Swagger (上篇)

Memcached LRU策略 (中篇)

Jmeter插件添加使用心得

网易云轻舟微服务斩获"创新产品奖"等两项大奖

知物由学 | AI在Facebook清理有害内容上扮演了什么角色?

直播

博客

网易公司版权所有 © 1997-2020 · 增值电信业务经营许可证B1-20180288 · 浙ICP备17006647号-2 · 杭州市滨江区网商路599号网易大厦 浙公网安备 33010802008718号