

给Swagger换了个新皮肤，瞬间高大上了！

原创 梦想de星空 macrozheng 2020-07-21 09:02

收录于合集

#mall学习教程（参考篇）

27个

Swagger作为一款API文档生成工具，虽然功能已经很完善了，但是还是有些不足的地方。偶然发现knife4j弥补了这些不足，赋予了Swagger更多的功能，今天我们来讲下它的使用方法。

knife4j简介

knife4j是springfox-swagger的增强UI实现，为Java开发者在使用Swagger的时候，提供了简洁、强大的接口文档体验。knife4j完全遵循了springfox-swagger中的使用方式，并在此基础上做了增强功能，如果你用过Swagger，你就可以无缝切换到knife4j。

快速开始

接下来我们来介绍下如何在SpringBoot中使用knife4j，仅需两步即可！

- 在pom.xml中增加knife4j的相关依赖；

```
<!--整合Knife4j-->
<dependency>
    <groupId>com.github.xiaoymin</groupId>
    <artifactId>knife4j-spring-boot-starter</artifactId>
    <version>2.0.4</version>
</dependency>
```

- 在Swagger2Config中增加一个@EnableKnife4j注解，该注解可以开启knife4j的增强功能；

```
/**
 * Swagger2API文档的配置
 */
```

```
@Configuration
@EnableSwagger2
@EnableKnife4j

public class Swagger2Config {

}
```

- 运行我们的SpringBoot应用，访问API文档地址即可查看：<http://localhost:8088/doc.html>



功能特点

接下来我们对比下Swagger，看看使用knife4j和它有啥不同之处！

JSON功能增强

平时一直使用Swagger，但是Swagger的JSON支持一直不是很好，JSON不能折叠，太长没法看，传JSON格式参数时，没有参数校验功能。这些痛点，在knife4j上都得到了解决。

- 返回结果集支持折叠，方便查看；



- 请求参数有JSON校验功能。



登录认证

knife4j也支持在头部添加Token，用于登录认证使用。

- 首先在 **Authorize** 功能中添加登录返回的Token；



- 之后在每个接口中就可以看到已经在请求头中携带了Token信息。



离线文档

knife4j支持导出离线文档，方便发送给别人，支持Markdown格式。

- 直接选择 **文档管理->离线文档** 功能，然后选择 **下载Markdown** 即可；



- 我们来查看下导出的Markdown离线文档，还是很详细的。



全局参数

knife4j支持临时设置全局参数，支持两种类型query(表单)、header(请求头)。

- 比如我们想要在所有请求头中加入一个参数appType来区分是android还是ios调用，可以在全局参数中添加；



- 此时再调用接口时，就会包含 **appType** 这个请求头了。



忽略参数属性

有时候我们创建和修改的接口会使用同一个对象作为请求参数，但是我们创建的时候并不需要id，而修改的时候会需要id，此时我们可以忽略id这个属性。

- 比如这里的创建商品接口，id、商品数量、商品评论数量都可以让后台接口生成无需传递，可以使用knife4j提供的 `@ApiOperationSupport` 注解来忽略这些属性；

```
/**
 * 品牌管理Controller
 * Created by macro on 2019/4/19.
 */
@Api(tags = "PmsBrandController", description = "商品品牌管理")
@Controller
@RequestMapping("/brand")
public class PmsBrandController {
    @Autowired
    private PmsBrandService brandService;

    private static final Logger LOGGER = LoggerFactory.getLogger(PmsBrandController.class);

    @ApiOperation("添加品牌")
    @ApiOperationSupport(ignoreParameters = {"id", "productCount", "productCommentCount"})
    @RequestMapping(value = "/create", method = RequestMethod.POST)
    @ResponseBody
    public CommonResult createBrand(@RequestBody PmsBrand pmsBrand) {
        CommonResult commonResult;
        int count = brandService.createBrand(pmsBrand);
        if (count == 1) {
            commonResult = CommonResult.success(pmsBrand);
            LOGGER.debug("createBrand success:{", pmsBrand);
        } else {
            commonResult = CommonResult.failed("操作失败");
            LOGGER.debug("createBrand failed:{", pmsBrand);
        }
        return commonResult;
    }
}
```

```
}  
}
```

- 此时查看接口文档时，发现这三个属性已经消失，这样前端开发查看接口文档时就不会觉得你定义无用参数了，是不很很好的功能！



参考资料

官方文档：<https://doc.xiaominfo.com/guide/>

项目源码地址

<https://github.com/macrozheng/mall-learning/tree/master/mall-tiny-knife4j>

推荐阅读

- [过来人聊聊经历，为什么不要再学JSP了！](#)
- [SpringBoot 2.3.0 新特性一览，快来跟我实践一波！](#)
- [给Swagger升级了新版本，没想到居然有这么多坑！](#)
- [fastjson到底做错了什么？为什么会被频繁爆出漏洞？](#)
- [微服务权限终极解决方案，Spring Cloud Gateway + Oauth2 实现统一认证和鉴权！](#)
- [听说你的JWT库用起来特别扭，推荐这款贼好用的！](#)
- [线上项目出BUG没法调试？推荐这款阿里开源的诊断神器！](#)

- [性能优越的轻量级日志收集工具，微软、亚马逊都在用！](#)
- [一个不容错过的Spring Cloud实战项目！](#)
- [我的Github开源项目，从0到20000 Star！](#)



欢迎关注，点个在看

收录于合集 #mall学习教程（参考篇） 27

上一篇

你居然还去服务器上捞日志，搭个日志收集系统难道不香么！

下一篇

Docker服务开放了这个端口，服务器分分钟变肉机！

阅读原文

喜欢此内容的人还喜欢

项目中到底该不该用Lombok?
macrozheng

