

Please visit website: <http://cxyroad.com>

Knife4j使用

=====

Knife4j和Swagger的区别

=====

Knife4j 是针对 Swagger 的增强解决方案，旨在为 Java 开发者提供更强大的 API 文档生成工具和更丰富的页面功能。Knife4j 在 Swagger 的基础上进行了扩展和优化，提供了更好的界面和更多的功能，以改善用户体验。

以下是 Knife4j 和 Swagger 的一些主要区别：

Swagger

- * Swagger 是一个开源的 API 开发工具套件，用于帮助开发者设计、构建、文档化和消费 RESTful Web 服务。
- * 它包括了 Swagger Editor (API 设计编辑器)、Swagger UI (API 文档页面)、Swagger Codegen (代码生成器) 等多个工具。
- * Swagger UI 提供了一个基本的 Web 界面，用户可以在其中查看 API 文档，并直接执行 API 请求进行测试。

Knife4j

- * Knife4j 是基于 Swagger 的一个增强工具，它提供了更加友好和美观的文档界面。
- * Knife4j 集成了 Swagger 原生的功能，并在此基础上增加了一些新特性，例如离线文档、接口调试等。
- * Knife4j 旨在提高开发者在使用 Swagger 时的体验，并提供一些额外的便利功能，比如个性化配置、接口文档的分组显示、全局参数配置等。
- * Knife4j 支持在微服务架构中使用，可以很好地与 Spring Cloud 集成。

Knife4j 是一个针对 Swagger 的增强解决方案，提供了更加友好的用户界面和一些额外的功能，但它本身并不定义如何描述API。Knife4j 使用 OpenAPI 规范（包括 `@Operation` 注解）作为文档的基础，并在此之上提供了更好的文档浏览和交互体验。简而言之，`@Operation` 注解用于定义API的行为，而 Knife4j 是一个展示和交互这些定义的工具。

当你在使用 Knife4j 时，你仍然会使用 `@Operation` 和其他 OpenAPI 注解来描述你的API，Knife4j 会读取这些注解生成的信息，并在其界面中以更加优化和用户友好的方式展示这些信息。例如，Knife4j 提供了文档搜索、接口测试、离线文档、动态参数等增强功能，这些都是基于你使用 OpenAPI 注解定义的 API 元数据之上的。

总结来说，`@Operation` 注解是用于定义API的工具，而 Knife4j 是用于展示和交互这些定义的工具。两者在API文档的生产和消费过程中都发挥着重要作用。

基础使用

=====

Knife4j 是针对 Swagger 的一个增强 UI 实现，用于更好地管理和使用 Swagger 生成的 API 文档。要在你的项目中使用 Knife4j，你可以按照以下步骤操作：

1. 添加 Knife4j 依赖

如果你的项目是基于 Spring Boot 的，你可以在 `pom.xml` 文件中添加 Knife4j 的依赖。以下是添加 Knife4j 依赖的例子：

```
...
<dependencies>
  <!-- 引入Knife4j -->
  <dependency>
    <groupId>com.github.xiaoymin</groupId>
    <artifactId>knife4j-spring-boot-starter</artifactId>
    <version>最新版本号</version>
  </dependency>
  <!-- Swagger的核心依赖 -->
  <dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger2</artifactId>
    <version>2.9.2</version>
  </dependency>
  <!-- Swagger UI的依赖，Knife4j已经包含了，所以不需要额外添加 -->
</dependencies>
...
```

请替换 `最新版本号` 为 Knife4j 当前的最新版本。

2. 配置 Swagger

配置 Swagger 来扫描你的 API 接口。在 Spring Boot 项目中，你可以添加一个配置类来启用 Swagger，并配置相关信息：

```
...  
import springfox.documentation.swagger2.annotations.EnableSwagger2;  
import com.github.xiaoymin.knife4j.spring.annotations.EnableKnife4j;  
  
@Configuration  
@EnableSwagger2  
@EnableKnife4j  
public class SwaggerConfig {  
    @Bean  
    public Docket createRestApi() {  
        return new Docket(DocumentationType.SWAGGER_2)  
            .apiInfo(apiInfo())  
            .select()  
            // 这里指定Controller扫描包路径  
            .apis(RequestHandlerSelectors.basePackage("com.example.controller"))  
            .paths(PathSelectors.any())  
            .build();  
    }  
  
    private ApiInfo apiInfo() {  
        return new ApiInfoBuilder()  
            .title("API 文档标题")  
            .description("API 文档描述")  
            .termsOfServiceUrl("服务条款URL")  
            .version("1.0")  
            .build();  
    }  
}  
...
```

3. 访问 Knife4j UI

启动你的 Spring Boot 应用程序后，你可以通过以下 URL 访问 Knife4j 提供的 UI 界面：

...

<http://localhost:8080/doc.html>

...

在 Knife4j 的 UI 界面中，你可以查看所有 API 文档的详细信息，并且可以直接在界面上测试 API 接口。

注意事项

- * Knife4j 主要是为了增强 Swagger 在 Spring Boot 项目中的使用体验，如果你的项目不是基于 Spring Boot，你可能需要参考 Knife4j 的其他集成方式。
- * 确保你的 API 接口类和方法使用了 Swagger 的注解，如 `@ApiOperation`、`@ApiParam`` 等，以便 Knife4j 能够正确生成 API 文档。
- * Knife4j 的版本更新可能会带来新的特性和改变，建议查看 Knife4j 的官方文档以获取最新的集成指南和使用说明。

通过使用 Knife4j，你可以获得比标准 Swagger UI 更丰富、更美观的 API 文档界面，提升开发者和最终用户的体验。

原文链接: <https://juejin.cn/post/7356078518096592896>