

2-5 利用Swagger搭建 REST API

140.143.132.225:8000/project-1/doc-17

开发前后端分离架构的项目，往往调试后端Web接口需要用到POSTMAN工具。虽然POSTMAN工具的功能非常强大，但是请求参数很多的情况下，我们手写这些参数和数据还是非常麻烦的。因此我们需要一个调试后端Web接口更加简便的方法。恰好Swagger提供了REST API调用方式，我们不需要借助任何工具的情况下，访问Swagger页面，就可以对Web接口进行调用和调试，这种调试方式的效率要远超POSTMAN软件。

一、添加依赖库

在 `pom.xml` 文件中添加Swagger依赖库，这里我们使用的是Swagger2版本，在UI方面，比Swagger1版本要好看很多。

```
1. <dependency>
2.     <groupId>io.springfox</groupId>
3.     <artifactId>springfox-swagger2</artifactId>
4.     <version>2.9.2</version>
5. </dependency>
6. <dependency>
7.     <groupId>io.springfox</groupId>
8.     <artifactId>springfox-swagger-ui</artifactId>
9.     <version>2.9.2</version>
10. </dependency>
```

二、创建Swagger配置类

在 `com.example.emos.wx.config` 包中创建 `SwaggerConfig` 类

```
1. package com.example.emos.wx.config;
2. import io.swagger.annotations.ApiOperation;
3. import org.springframework.context.annotation.Bean;
4. import org.springframework.context.annotation.Configuration;
5. import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
6. import springfox.documentation.builders.ApiInfoBuilder;
7. import springfox.documentation.builders.PathSelectors;
8. import springfox.documentation.builders.RequestHandlerSelectors;
9. import springfox.documentation.service.ApiInfo;
10. import springfox.documentation.service.ApiKey;
11. import springfox.documentation.service.AuthorizationScope;
12. import springfox.documentation.service.SecurityReference;
13. import springfox.documentation.spi.DocumentationType;
14. import springfox.documentation.spring.web.plugins.ApiSelectorBuilder;
15. import springfox.documentation.spring.web.plugins.Docket;
16. import springfox.documentation.swagger2.annotations.EnableSwagger2;
17. import java.util.ArrayList;
18. import java.util.List;
19. @Configuration
20. @EnableSwagger2
21. public class SwaggerConfig {
22.     @Bean
23.     public Docket createRestApi() {
24.         Docket docket = new Docket(DocumentationType.SWAGGER_2);
25.         // ApiInfoBuilder 用于在Swagger界面上添加各种信息
26.         ApiInfoBuilder builder = new ApiInfoBuilder();
27.         builder.title("EMOS在线办公系统");
28.         ApiInfo apiInfo = builder.build();
29.         docket.apiInfo(apiInfo);
30.         // ApiSelectorBuilder 用来设置哪些类中的方法会生成到REST API中
31.         ApiSelectorBuilder selectorBuilder = docket.select();
32.         selectorBuilder.paths(PathSelectors.any()); //所有包下的类
33.         //使用@ApiOperation的方法会被提取到REST API中
34.
35.         selectorBuilder.apis(RequestHandlerSelectors.withMethodAnnotation(ApiOperation.class));
36.         docket = selectorBuilder.build();
37.         /*
38.          * 下面的语句是开启对JWT的支持，当用户用Swagger调用受JWT认证保护的方法，
39.          * 必须要先提交参数（例如令牌）
40.          */
41.         //存储用户必须提交的参数
42.         List<ApiKey> apikey = new ArrayList();
43.         //规定用户需要输入什么参数
44.         apikey.add(new ApiKey("token", "token", "header"));
45.         docket.securitySchemes(apikey);
46.         //如果用户JWT认证通过，则在Swagger中全局有效
47.         AuthorizationScope scope = new AuthorizationScope("global", "accessEverything");
48.         AuthorizationScope[] scopeArray = {scope};
49.         //存储令牌和作用域
50.         SecurityReference reference = new SecurityReference("token", scopeArray);
51.         List refList = new ArrayList();
52.         refList.add(reference);
53.         SecurityContext context =
54.             SecurityContext.builder().securityReferences(refList).build();
```

```
54.         List cxtList = new ArrayList();
55.         cxtList.add(context);
56.         docket.securityContexts(cxtList);
57.         return docket;
58.     }
59. }
```

三、编写测试Web接口

在 `com.example.emos.wx.controller` 包中创建 `TestController` 类。

```
1. package com.example.emos.wx.controller;
2. import com.example.emos.wx.common.util.R;
3. import io.swagger.annotations.Api;
4. import io.swagger.annotations.ApiOperation;
5. import org.springframework.web.bind.annotation.GetMapping;
6. import org.springframework.web.bind.annotation.RequestMapping;
7. import org.springframework.web.bind.annotation.RestController;
8. @RestController
9. @RequestMapping("/test")
10. @Api("测试Web接口")
11. public class TestController {
12.     @GetMapping("/sayHello")
13.     @ApiOperation("最简单的测试方法")
14.     public R sayHello(){
15.         return R.ok().put("message", "HelloWorld");
16.     }
17. }
```



四、测试Web接口

打开浏览器，访问 `http://127.0.0.1:8080/emos-wx-api/swagger-ui.html`

EMOS在线办公系统

[Base URL: 127.0.0.1:8080/emos-wx-api]

<http://127.0.0.1:8080/emos-wx-api/v2/api-docs>

Authorize



test-controller Test Controller



GET

/test/sayHello 最简单的测试方法

