**Spring Boot 整合之Swagger2**

# 1 引入依赖

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version>2.8.0</version>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger-ui</artifactId>

<version>2.8.0</version>

</dependency>

并在主程序中加上：

@EnableSwagger2

# 2 Controller层的swagger注解

**Swagger通过注解表明该接口会生成文档，包括接口名、请求方法、参数、返回信息的等等。**

**2.1 @Api：修饰整个类，描述Controller的作用**

用在请求的类上，表示对类的说明

* + tags="说明该类的作用，可以在UI界面上看到的注解"
  + value="该参数没什么意义，在UI界面上也看到，所以不需要配置"

如: @Api(value="/test1", tags="测试接口模块")

**2.2 @ApiOperation：描述一个类的一个方法，或者说一个接口**

用在请求的方法上，说明方法的用途、作用

* + value="说明方法的用途、作用"
  + notes="方法的备注说明"

如: @ApiOperation(value="添加用户信息", nodes="添加用户信息")

**2.3 @ApiParam：单个参数描述**

如

public xxx edit(

@ApiParam(name = "title", value = "公告标题", required = true) @RequestParam("title") String title,

@ApiParam(name = "content", value = "公告内容", required = true) @RequestParam("content") String content){

**2.4 @ApiResponses：HTTP响应整体描述**

**@ApiResponse：HTTP响应其中1个描述**

@ApiResponse：用在@ApiResponses中，一般用于表达一个错误的响应信息

code：数字，例如400

message：信息，例如"请求参数没填好"

response：抛出异常的类

**2.5 @ApiImplicitParams：多个请求参数**

**@ApiImplicitParam：一个请求参数**

@ApiImplicitParams：用在请求的方法上，表示一组参数说明

@ApiImplicitParam：用在@ApiImplicitParams注解中，指定一个请求参数的各个方面

name：参数名

value：参数的汉字说明、解释

required：参数是否必须传

paramType：参数放在哪个地方

· header --> 请求参数的获取：@RequestHeader

· query --> 请求参数的获取：@RequestParam

· path（用于restful接口）--> 请求参数的获取：@PathVariable

· body（不常用）

· form（不常用）

dataType：参数类型，默认String，其它值dataType="Integer"

defaultValue：参数的默认值

# 3 访问 SpringBoot集成的Swagger2

<http://127.0.0.1/swagger-ui.html>

# 4问题

**No mapping found for HTTP request with URI [/swagger-ui.html]**

解决办法:

@Configuration

@EnableWebMvc

public class WebMvcConfigurerConfig implements WebMvcConfigurer {

@Override

public void addResourceHandlers(ResourceHandlerRegistry registry) {

// Swagger

registry.addResourceHandler("swagger-ui.html")

.addResourceLocations("classpath:/META-INF/resources/");

registry.addResourceHandler("/webjars/\*\*")

.addResourceLocations("classpath:/META-INF/resources/webjars/");

}

}