swagger用于定义API文档。

**1. 好处：**

* 前后端分离开发
* API文档非常明确
* 测试的时候不需要再使用URL输入浏览器的方式来访问Controller
* 传统的输入URL的测试方式对于post请求的传参比较麻烦（当然，可以使用postman这样的浏览器插件）
* springfox基于swagger2，兼容老版本

**2. maven引入：**

<!--springfox--><**dependency**>

<**groupId**>com.fasterxml.jackson.core</**groupId**>

<**artifactId**>jackson-databind</**artifactId**>

<**version**>2.4.4</**version**></**dependency**><**dependency**>

<**groupId**>io.springfox</**groupId**>

<**artifactId**>springfox-swagger2</**artifactId**>

<**version**>2.5.0</**version**></**dependency**>

**3. 引入bean**

@Configuration@EnableSwagger2@EnableWebMvc@ComponentScan(basePackages ={**"扫描类"**})

**public class** SwaggerConfig{

/\*\* \* Every Docket bean is picked up by the swagger-mvc framework - allowing for multiple \* swagger groups i.e. same code base multiple swagger resource listings. \*/ @Bean **public** Docket customDocket(){

**return new** Docket(DocumentationType.SWAGGER\_2);

//此处还有很多其他选项，包括路径过滤，api说明等

}}

说明：

* 引入了一个注解@EnableSwagger2来启动swagger注解。（启动该注解使得用在controller中的swagger注解生效，覆盖的范围由@ComponentScan的配置来指定，这里默认指定为根路径"com.xxx.firstboot"下的所有controller）
* 非springboot框架需要引入注解@EnableWebMvc
* 在springmvc配置文件中添加bean

**<bean** class="cn.conf.SwaggerConfig"**/>**

**4. 展示：**

从github(https://github.com/wordnik/swagger-ui)上下载Swagger-UI, 把该项目dist目录下的内容拷贝到项目：

修改index.html    url = "/v2/api-docs";

指定mvc映射，让外部能访问到dist目录下的index.html,如：

在springmvc中增加    <mvc:resources mapping="/dist/\*\*" location="/WEB-INF/dist/"/>配置

**5.常用注解：**

* @Api：用在类上，说明该类的作用
* @ApiOperation：用在方法上，说明方法的作用
* @ApiImplicitParams：用在方法上包含一组参数说明
* @ApiImplicitParam：用在@ApiImplicitParams注解中，指定一个请求参数的各个方面
  + 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）
  + name：参数名
  + dataType：参数类型
  + required：参数是否必须传
  + value：参数的意思
  + defaultValue：参数的默认值
* @ApiResponses：用于表示一组响应
* @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
* @ApiModel：描述一个Model的信息（这种一般用在post创建的时候，使用@RequestBody这样的场景，请求参数无法使用@ApiImplicitParam注解进行描述的时候）
  + @ApiModelProperty：描述一个model的属性

注意：

在controller 使用springfox的  
@ApiParam(name="queryCondition", value="查询条件", required=true) @RequestBody QueryCondition queryCondition  
注解，可在页面看到QueryCondition类型参数  
  
  
使用@ApiOperation(value = "Greeting by Name",    
            notes = "Say hello to the people",    
            response = SayingRepresentation.class,    
            position = 0)  
可以看到返回值类型

**6.[测试](http://lib.csdn.net/base/softwaretest" \o "软件测试知识库" \t "http://blog.csdn.net/cws1214/article/details/_blank)：**

启动服务，浏览器输入"http://localhost:8080/dist/index.html"