## 添加jar

|  |
| --- |
| <swagger.version>2.6.1</swagger.version>  <jackson.version>2.9.2</jackson.version>  <dependency>  <groupId>io.springfox</groupId>  <artifactId>springfox-swagger2</artifactId>  <!-- <version>${swagger.version}</version> -->  </dependency>  <dependency>  <groupId>io.springfox</groupId>  <artifactId>springfox-swagger-ui</artifactId>  <!-- <version>${swagger.version}</version> -->  </dependency>    <!-- jackson-databind 依赖于jackson-core  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-core</artifactId>  </dependency> -->  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-databind</artifactId>  <!-- <version>2.9.2</version> -->  </dependency>  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-annotations</artifactId>  <!-- <version>2.9.2</version> -->  </dependency> |

## 配置spring-mvc.xml支持swagger

|  |
| --- |
| <!-- 添加的效果：value没起作用，反而是description 起作用 -->  <bean class=*"springfox.documentation.swagger2.configuration.Swagger2DocumentationConfiguration"* id=*"swagger2Config"*/>    <!-- 配置哪些是静态资源，缺省Servlet 直接返回 -->  <mvc:resources mapping=*"swagger-ui.html"* location=*"classpath:/META-INF/resources/"*/>  <mvc:resources mapping=*"/webjars/\*\*"* location=*"classpath:/META-INF/resources/webjars/"*/> |

还是建议屏蔽这个bean，新建自定义的bean

|  |
| --- |
| **package** com.shuframework.config;  **import** org.springframework.context.annotation.Bean;  **import** org.springframework.context.annotation.ComponentScan;  **import** org.springframework.context.annotation.Configuration;  **import** org.springframework.web.servlet.config.annotation.EnableWebMvc;  **import** org.springframework.web.servlet.config.annotation.WebMvcConfigurationSupport;  **import** springfox.documentation.builders.ApiInfoBuilder;  **import** springfox.documentation.builders.PathSelectors;  **import** springfox.documentation.builders.RequestHandlerSelectors;  **import** springfox.documentation.service.ApiInfo;  **import** springfox.documentation.service.Contact;  **import** springfox.documentation.spi.DocumentationType;  **import** springfox.documentation.spring.web.plugins.Docket;  **import** springfox.documentation.swagger2.annotations.EnableSwagger2;    /\*\*  \* 描述：swagger2配置类  \*/  //@EnableWebMvc  //@ComponentScan(basePackages={"com.shuframework"}) //需要扫描的包路径  @EnableSwagger2  @Configuration  **public** **class** SwaggerApiConfig{  // public class SwaggerApiConfig extends WebMvcConfigurationSupport{  @Bean  **public** Docket createRestApi() {  **return** **new** Docket(DocumentationType.***SWAGGER\_2***)  .apiInfo(apiInfo())  .select()  .apis(RequestHandlerSelectors.*basePackage*("com.shuframework"))  .paths(PathSelectors.*any*())  .build();  }    **private** ApiInfo apiInfo() {  **return** **new** ApiInfoBuilder()  .title("ssm-api接口")  .termsOfServiceUrl("")  // .contact(new Contact("我本码农", "", "15851503917@163.com"))  .version("v1.0")  .build();  }  } |

这个bean如果是扫描的，需要注意@Configuration 是基于@Component的，则在扫描时需要将包根路径引入。

如：<context:component-scan base-package=*"com.shuframework.config"* />

## 配置controller说明

|  |
| --- |
| 常用注解使用说明介绍，请结合springfox使用查看  @ApiIgnore 忽略注解标注的类或者方法，不添加到API文档中  @ApiOperation 展示每个API基本信息  value api名称  notes 备注说明  @ApiImplicitParam 用于规定接收参数类型、名称、是否必须等信息  name 对应方法中接收参数名称  value 备注说明  required 是否必须 boolean  paramType 参数类型 body、path、query、header、form中的一种  body 使用@RequestBody接收数据 POST有效  path 在url中配置{}的参数  query 普通查询参数 例如 ?query=q ,jquery ajax中data设置的值也可以，例如 {query:”q”},springMVC中不需要添加注解接收  header 使用@RequestHeader接收数据  form 笔者未使用，请查看官方API文档  dataType 数据类型，如果类型名称相同，请指定全路径，例如 dataType = “java.util.Date”，springfox会自动根据类型生成模型  @ApiImplicitParams 包含多个@ApiImplicitParam  @ApiModelProperty 对模型中属性添加说明，例如 上面的PageInfoBeen、BlogArticleBeen这两个类中使用，只能使用在类中。  value 参数名称  required 是否必须 boolean  hidden 是否隐藏 boolean  其他信息和上面同名属性作用相同，hidden属性对于集合不能隐藏，目前不知道原因  @ApiParam 对单独某个参数进行说明，使用在类中或者controller方法中都可以。注解中的属性和上面列出的同名属性作用相同 |

例子：

|  |
| --- |
| @RequestMapping("/list")  @ResponseBody  @ApiOperation(value = "分页查询", notes = "aa", httpMethod = "POST", produces = MediaType.***APPLICATION\_JSON\_VALUE***)  //该种方式默认是paramType = "body"，application/json，一般不存在多个，不同paramType可以存在一个  @ApiImplicitParams({  @ApiImplicitParam(name = "pageIndex", value = "当前页数", required = **true**, paramType = "query", dataType = "integer"),  @ApiImplicitParam(name = "pageSize", value = "条数", required = **true**, paramType = "query", dataType = "integer")  })  **public** Result list(**int** pageIndex, **int** pageSize) {  Result result = **new** Result();  //第一页默认为 1  Page<SysUser> page = **new** Page<>(pageIndex, pageSize);  page = userService.selectPage(page);    result.setSuccess(**true**);  result.setCode(20000);  result.setData(page);  **return** result;  } |

如果配置了 @ApiImplicitParam 请求参数说明，paramType = "query", 配置为body不一定拿得到。

最后启动项目，**如：http://localhost:8080/ssm-init/swagger-ui.html#**

