



《Spring Boot 基础教程》

第 30 节

Spring Boot 集成 Swagger

一、Swagger 是什么？

Swagger 是一个规范和完整的框架，用于生成、描述、调用和可视化 RESTful 风格的 Web 服务。

<http://swagger.io/>

Springfox 的前身是 swagger-springmvc，是一个开源的 API doc 框架，可以将我们的 Controller 的方法以文档的形式展现，基于 Swagger。

<http://springfox.github.io/springfox/>

二、添加 jar

```
<!-- Swagger -->
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger-ui</artifactId>
    <version>2.6.0</version>
</dependency>

<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger2</artifactId>
    <version>2.6.0</version>
</dependency>
```

三、配置

```
import static springfox.documentation.builders.PathSelectors.regex;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import springfox.documentation.builders.ApiInfoBuilder;
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;

/**
 * SwaggerConfig
 */
@Configuration
@EnableSwagger2
public class Swagger2Configuration {
```



```
/**
 *
 * @return
 */
@Bean
public Docket accessToken() {
    return new Docket(DocumentationType.SWAGGER_2).groupName("api")// 定义组
        .select() // 选择那些路径和 api 会生成 document

    .apis(RequestHandlerSelectors.basePackage("com.roncoo.example.controller")) // 拦截的包
    路径

        .paths(regex("/api/.*/"))// 拦截的接口路径
        .build() // 创建
        .apiInfo(apiInfo()); // 配置说明
}

private ApiInfo apiInfo() {
    return new ApiInfoBuilder()//
        .title("龙果学院")// 标题
        .description("spring boot 全集")// 描述
        .termsOfServiceUrl("http://www.roncoo.com")//
        .contact(new Contact("wujing", "http://www.roncoo.com",
"297115770@qq.com"))// 联系
        // .license("Apache License Version 2.0")// 开源协议

        // .licenseUrl("https://github.com/springfox/springfox/blob/master/LICENSE")// 地址
        .version("1.0")// 版本
        .build();
}
}
```

四、测试

<http://localhost:8080/swagger-ui.html>

五、自定义(注解的使用)

```
@ApiIgnore
    忽略暴露的 api

@ApiOperation(value = "查找", notes = "根据用户 ID 查找用户")
    添加说明
```

其他注解:

@Api: 用在类上, 说明该类的作用
@ApiImplicitParams: 用在方法上包含一组参数说明
@ApiResponses: 用于表示一组响应



@ApiResponse: 用在@ApiResponse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

更多请查看

<https://github.com/swagger-api/swagger-core/wiki/Annotations>

更多课程信息，请关注 龙果学院 官方网站 <http://www.roncoo.com/>

或关注 龙果 微信公众号 RonCoo_com

