# Swagger学习笔记

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。总体目标是使客户端和文件系统作为服务器以同样的速度来更新。文件的方法，参数和模型紧密集成到服务器端的代码，允许API来始终保持同步。

然而，手动编写此类文档并在代码更改时保持更新是不现实的。这就是SpringFox发挥作用的地方。它是Spring Framework的Swagger集成。它可以自动检查您的类，检测控制器，它们的方法，它们使用的模型类以及它们映射到的URL。没有任何手写文档，只需检查应用程序中的类，它就可以生成大量有关API的信息。

作用：

1.接口的文档在线自动生成

2.功能测试

## maven导入Swagger

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version>2.6.1</version>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

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

<version>2.6.1</version>

</dependency>

## 基本配置

添加依赖项后，需要提供一些基本的spring配置。虽然您可以在技术上使用现有配置文件之一，但最好为其配置单独的文件。

**在Application.java同级创建Swagger2的配置类Swagger2（可随意命名）**

/\*\*

\* Swagger2配置类

\* 在与spring boot集成时，放在与Application.java同级的目录下。

\* 通过@Configuration注解，让Spring来加载该类配置。

\* 再通过@EnableSwagger2注解来启用Swagger2。

\*/

@Configuration

@EnableSwagger2

public class Swagger2 {

/\*\*

\* 创建API应用

\* apiInfo() 增加API相关信息

\* 通过select()函数返回一个ApiSelectorBuilder实例,用来控制哪些接口暴露给Swagger来展现，

\* 本例采用指定扫描的包路径来定义指定要建立API的目录。

\*

\* @return

\*/

@Bean

public Docket createRestApi() {

return new Docket(DocumentationType.SWAGGER\_2)

.apiInfo(apiInfo())

.select()

.apis(RequestHandlerSelectors.basePackage("com.swaggerTest.controller"))

.paths(PathSelectors.any())

.build();

}

/\*\*

\* 创建该API的基本信息（这些基本信息会展现在文档页面中）

\* 访问地址：http://项目实际地址/swagger-ui.html

\* @return

\*/

private ApiInfo apiInfo() {

return new ApiInfoBuilder()

.title("Spring Boot中使用Swagger2构建RESTful APIs")

.description("更多请关注http://www.baidu.com")

.termsOfServiceUrl("http://www.baidu.com")

.contact("sunf")

.version("1.0")

.build();

}

}

创建Docket的Bean之后，apiInfo()用来创建该Api的基本信息。

完成上述配置后，已经可以生产文档内容了，但是这样的文档主要针对请求本身，描述的主要来源是函数的命名，对用户并不友好，我们通常需要自己增加一些说明来丰富文档内容。

**Swagger使用的注解及其说明：**

@Api ：用在类上，说明该类的作用。

@ApiOperation ：注解来给API增加方法说明。

@ApiImplicitParams ：用在方法上包含一组参数说明。

@ApiImplicitParam ：用来注解来给方法入参增加说明。

@ApiResponses ：用来表示一组响应。

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

@ApiModel ：描述一个Model信息（一般用在请求参数无法使用@ApiImplicitParam注解进行描述的时候）

