

# Swagger生成优美文档

## 1.0 引入Pom文件

```
<!-- swagger -->
<!-- 之所以要排除，是因为如果不排除会报NumberFormatException的警告。 -->
<!-- 参考: https://github.com/springfox/springfox/issues/2265-->
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger2</artifactId>
    <version>2.9.2</version>
    <exclusions>
        <exclusion>
            <groupId>io.swagger</groupId>
            <artifactId>swagger-annotations</artifactId>
        </exclusion>
        <exclusion>
            <groupId>io.swagger</groupId>
            <artifactId>swagger-models</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger-ui</artifactId>
    <version>2.9.2</version>
</dependency>
<dependency>
    <groupId>io.swagger</groupId>
    <artifactId>swagger-annotations</artifactId>
    <version>1.5.21</version>
</dependency>
<dependency>
    <groupId>io.swagger</groupId>
    <artifactId>swagger-models</artifactId>
    <version>1.5.21</version>
</dependency>
```

## 2.0 引入mavenPlugins

```
<!--          springfox2markup-maven-plugin 插件的作用是读取
http://localhost:8080/v2/api-docs 的信息，生成ASCIIDOC文档。当然你也可以生成其他格式，比如
Markdown等等。-->
<!--          这款插件还有很多使用姿势，详见
https://github.com/Swagger2Markup/swagger2markup-maven-plugin-->
<plugin>
    <groupId>io.github.swagger2markup</groupId>
    <artifactId>swagger2markup-maven-plugin</artifactId>
```

```

<version>1.3.1</version>
<configuration>
    <!-- api-docs访问url -->
    <swaggerInput>http://localhost:8086/v2/api-docs</swaggerInput>
    <!-- 生成为单个文档，输出路径 -->
    <outputFile>src/docs/asciidoc/generated/all</outputFile>
    <config>
        <!-- ascii格式文档 -->

<swagger2markup.markupLanguage>ASCIIDOC</swagger2markup.markupLanguage>

<swagger2markup.pathsGroupedBy>TAGS</swagger2markup.pathsGroupedBy>
    </config>
</configuration>
</plugin>

<!--          asciidoctor-maven-plugin 插件同样也有很多姿势，详见：
https://github.com/asciidoctor/asciidoctor-maven-plugin-->
<!--          生成的文件在 src/docs/asciidoc/html -->
<plugin>
    <groupId>org.asciidoctor</groupId>
    <artifactId>asciidoctor-maven-plugin</artifactId>
    <version>1.5.6</version>
    <configuration>
        <!-- asciidoc文档输入路径 -->
        <sourceDirectory>src/docs/asciidoc/generated</sourceDirectory>
        <!-- html文档输出路径 -->
        <outputDirectory>src/docs/asciidoc/html</outputDirectory>
        <backend>html</backend>
        <sourceHighlighter>coderay</sourceHighlighter>
        <!-- html文档格式参数 -->
        <attributes>
            <doctype>book</doctype>
            <toc>left</toc>
            <toclevels>3</toclevels>
            <numbered></numbered>
            <hardbreaks></hardbreaks>
            <sectlinks></sectlinks>
            <sectanchors></sectanchors>
        </attributes>
    </configuration>
</plugin>

```

## 3.0 swagger配置类

```

//类似于swagger配置类
@EnableSwagger2
@Configuration
public class SwaggerConfiguration {
    private ApiInfo apiInfo() {
        return new ApiInfo(
            "share-app Swagger文档",
            "github地址:www.baidu.com",
            "API V1.0",
            "Terms of service",

```

```

        new Contact("王林", "https://wl.cn", "1987153442@qq.com"),
        "Apache", "http://www.apache.org/", Collections.emptyList()
    );
}

@Bean
public Docket api(){
    return new Docket(DocumentationType.SWAGGER_2)
        .select()
        .apis(RequestHandlerSelectors.basePackage("com.wl.contentcenter"))
        .build()
        .apiInfo(apiInfo());
}
}

```

## 4.0在实体类需要加上注解 类似于Swagger上注解 具体可以查看

<https://juejin.im/post/6844904069924929550>

```

@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
@ApiModel("搜索Dto")
public class SearchDto {
    @ApiModelProperty(name = "searchKey", value = "搜索关键词")
    private String searchKey;
}

```

## 5.0 controller上引入注解

```

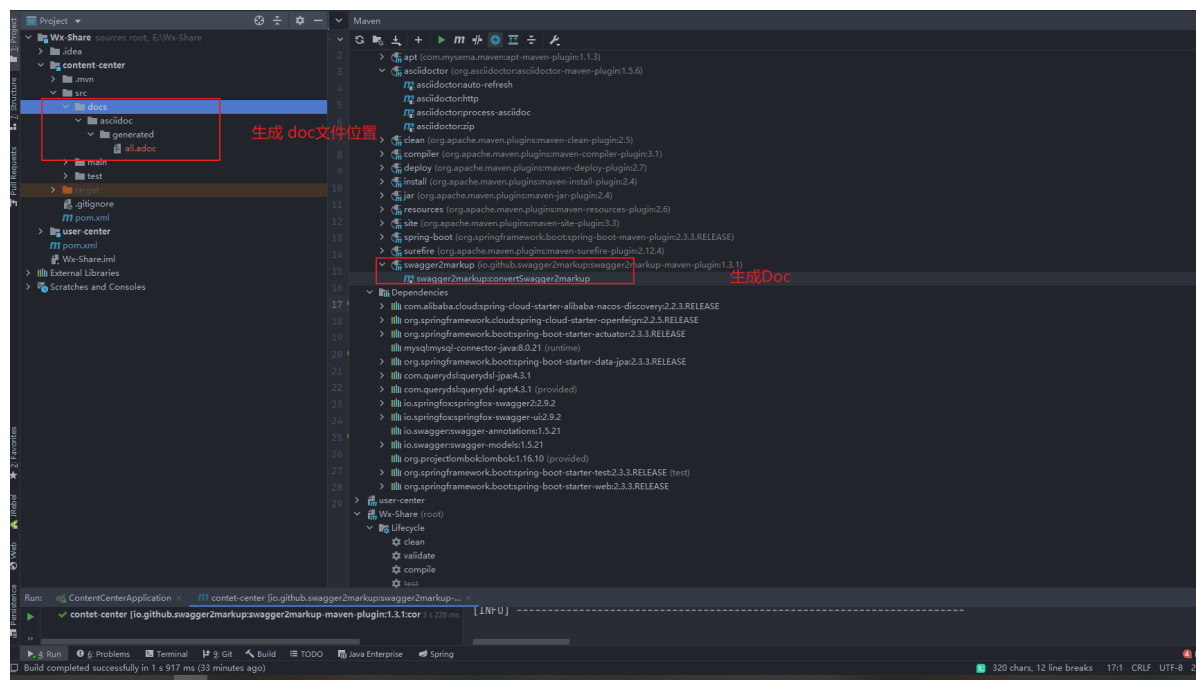
@RestController
@RequestMapping("/notice")
//swagger注解
@Api(tags = "公告接口", value = "公告相关的Api")
public class NoticeController {
    @Resource private NoticeService noticeService;

    @GetMapping("/findLast")
    //swagger注解
    @ApiOperation(value = "公告", notes = "查询公告")
    public NoticeVo findLast() {
        return noticeService.findLatestNotice();
    }
}

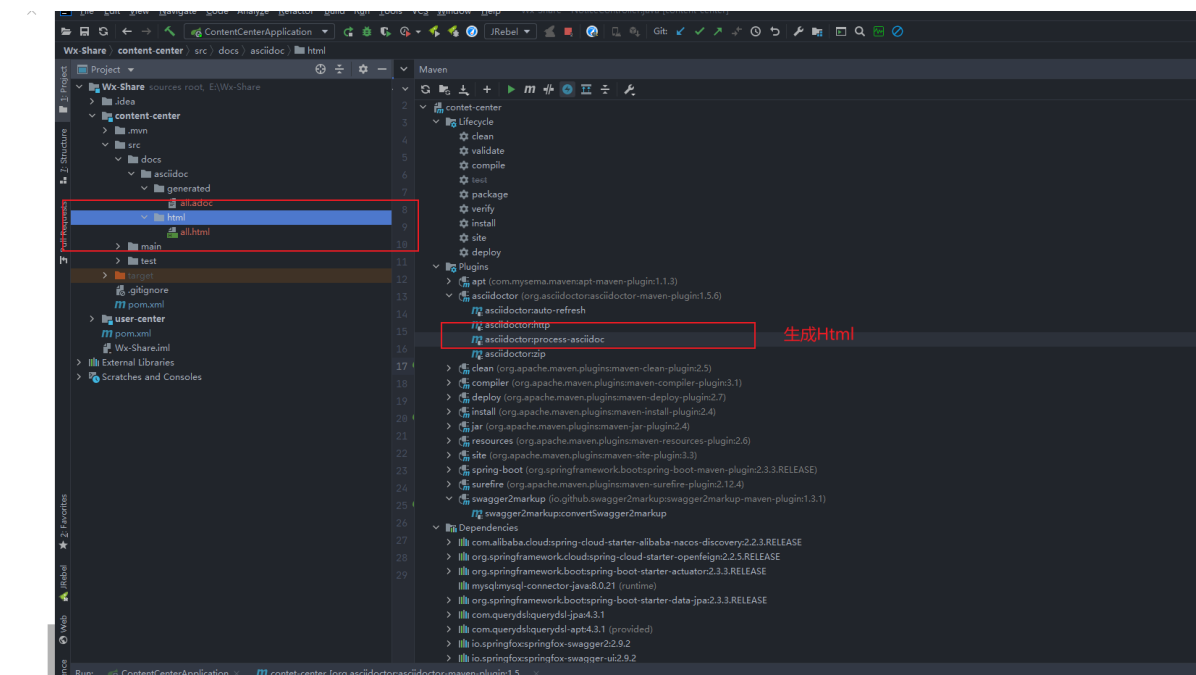
```

## 6.0 mavenPlugin操作

### 第一步生成Doc文件



### 第二部生成Html文件



## 7.0文档最终效果如图所示

## Table of Contents

- 1. Overview
  - 1.1. Version information
  - 1.2. Contact information
  - 1.3. License information
  - 1.4. URI scheme
  - 1.5. Tags
- 2. Resources
  - 2.1. Content-controller
    - 2.1.1. getInstance
  - 2.2. 公告接口
    - 2.2.1. 公告
  - 2.3. 分享接口
    - 2.3.1. 投稿
    - 2.3.2. 编辑接口
    - 2.3.3. 查询分享
    - 2.3.4. 查询分享
    - 2.3.5. 搜索分享
- 3. Definitions
  - 3.1. RestResult
  - 3.2. Timestamp
  - 3.3. 公告Vo
  - 3.4. 分享Dto
  - 3.5. 分享Vo
  - 3.6. 投稿Dto
  - 3.7. 搜索Dto
  - 3.8. 编辑投稿Dto

# share-app Swagger文档

## 1. Overview

github地址: [www.baidu.com](http://www.baidu.com)

### 1.1. Version information

Version : API V1.0

### 1.2. Contact information

Contact : 王林

Contact Email : [1987153442@qq.com](mailto:1987153442@qq.com)

### 1.3. License information

License : Apache

License URL : <http://www.apache.org/>

Terms of service : Terms of service

### 1.4. URI scheme

Host : localhost:8086

BasePath : /

### 1.5. Tags

- content-controller : Content Controller
- 公告接口 : Notice Controller
- 分享接口 : Share Controller

## 8.0总结

swagger总结 在swagger基础上引入doc文档 引入maven插件 以此生成文档

当然我们也可以自己封装jar包来满足我们日常需求

//springboot项目重复代码 提成jar包案例

<https://github.com/purgeteam/unified-dispose-springboot>