

Spring Boot 集成 Swagger，再也不写接口文档了！

原创：栈长 Java技术栈 今天



之前的文章介绍了《[推荐一款接口 API 设计神器！](#)》，今天栈长给大家介绍下如何与优秀的 Spring Boot 框架进行集成，简直不能太简单。

你所需具备的基础

- [告诉你，Spring Boot 真是个牛逼货！](#)
- [Spring Boot 核心配置文件详解](#)
- [Spring Boot 开启的 2 种方式](#)
- [Spring Boot 自动配置原理、实战](#)
- [Spring Boot 2.x 启动全过程源码分析](#)

更多请在Java技术栈微信公众号后台回复关键字：**boot**。

Spring Boot 集成 Swagger

1、添加依赖

Maven依赖示例：

```
<!-- Swagger -->
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger2</artifactId>
</dependency>
<dependency>
    <groupId>io.springfox</groupId>
```

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

2、在 Spring Boot 配置文件中添加配置参数。

```
swagger:
  title: API标题
  description: API描述
  version: 1.0
  terms-of-service-url: http://www.javastack.cn/
  base-package: cn.javastack.test.web
  contact:
    name: Javastack
    url: http://www.javastack.cn/
    email: admin@javastack.cn
```

3、添加配置类

```
@Getter
@Setter
@Configuration
@EnableSwagger2
@ConditionalOnClass(EnableSwagger2.class)
@ConfigurationProperties(prefix = "swagger")
public class SwaggerConfig {

    /**
     * API接口包路径
     */
    private String basePackage;

    /**
     * API页面标题
     */
}
```

```
private String title;

/**
 * API描述
 */
private String description;

/**
 * 服务条款地址
 */
private String termsOfServiceUrl;

/**
 * 版本号
 */
private String version;

/**
 * 联系人
 */
private Contact contact;

@Bean
public Docket api() {
    return new Docket(DocumentationType.SWAGGER_2)
        .apiInfo(apiInfo())
        .select()
        .apis(RequestHandlerSelectors.basePackage(basePackage))
        .paths(PathSelectors.any())
        .build();
}

private ApiInfo apiInfo() {
    return new ApiInfoBuilder()
        .title(title)
        .description(description)
        .termsOfServiceUrl(termsOfServiceUrl)
        .version(version)
        .contact(contact)
        .build();
}
```

```
}  
  
}
```

如何使用

Swagger 默认会根据配置的包，扫描所有接口并生成对应的 API 描述和参数信息，但这样不是很直观，需要对每个接口和参数进行自定义描述。

常用的 Swagger 注解如下。

| 注解名称 | 使用说明 |
|--------------------|-------------|
| @Api | 描述一个 API 类 |
| @ApiImplicitParam | 描述一个请求参数 |
| @ApiImplicitParams | 描述一组请求参数 |
| @ApiModel | 描述一个返回的对象 |
| @ApiModelProperty | 描述一个返回的对象参数 |
| @ApiOperation | 描述一个 API 方法 |
| @ApiParam | 描述一个方法的参数 |
| @ApiResponse | 描述一个请求响应 |

| 注解名称 | 使用说明 |
|----------------------------|----------|
| <code>@ApiResponses</code> | 描述一组请求响应 |

使用示例如：

```
@Api(description = "登录模块")
@RestController
public class LoginController {

    @ApiOperation(value = "登录", httpMethod = "POST")
    @ApiImplicitParams({
        @ApiImplicitParam(name = "username", value = "用户名", dataType = "string", paramType = "query"),
        @ApiImplicitParam(name = "password", value = "密码", dataType = "string", paramType = "query")})
    @PostMapping(value = "/login")
    public Object login(@RequestParam("username") String username, @RequestParam("password") String password) {

        // ...

    }
}
```

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

打开 swagger-ui 界面，可以看到所有的 API 接口定义，也可以在上面发起接口测试。

关注Java技术栈微信公众号，在后台回复：**工具**，获取栈长整理的更多的工具绝技，都是实战干货，以下仅为部分预览。

- Java 开发必知道的国外 10 大网站
- 免费在线创作流程图、思维导图软件
- 推荐一款代码神器，代码量至少省一半！

- 推荐一款接口 API 设计神器！
- 超详细的 Git 实战教程，傻瓜一看也会！
-

最近干货分享

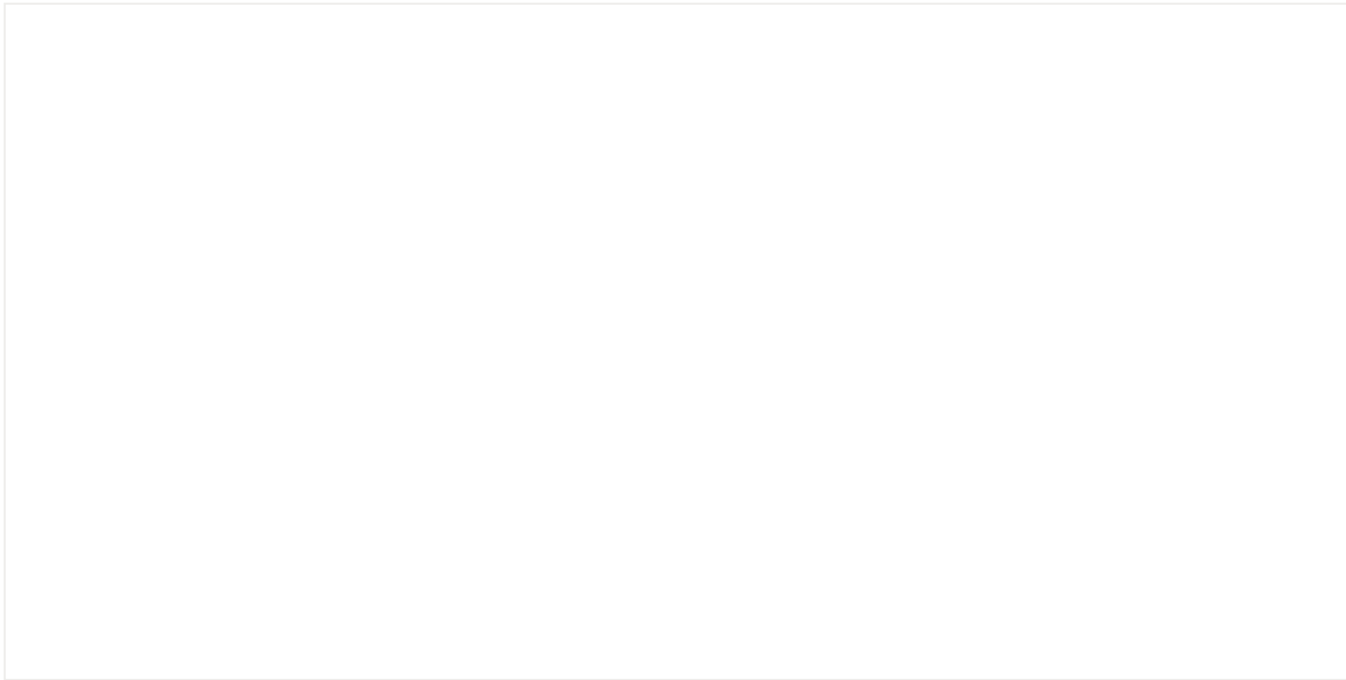
[为什么面试你要25K，HR只给你20K？](#)

[Spring 系列干货，2019最新整理版！](#)

[你真的搞懂 transient 关键字了吗？](#)

[推荐一个技术圈子，30k的offer靠它了！](#)

[分享一份Java架构师学习资料！](#)



点击「[阅读原文](#)」和栈长学更多...

[阅读原文](#)