# 什么是swagger?

**swagger是一个自动生成接口文档的工具；**

# 为什么使用swagger？

前后端分离的项目，接口文档的存在十分重要。与手动编写接口文档不同，swagger是一个自动生成接口文档的工具，在需求不断变更的环境下，手动编写文档的效率实在太低。

# Springboot如何整合swagger？

## 1.引入依赖

**Maven方式:**

<dependency>

<groupId>com.spring4all</groupId>

<artifactId>swagger-spring-boot-starter</artifactId>

<version>2.0.0.RELEASE</version>

</dependency>

## 2. Swagger3Config 配置

Docket.enable(true):是否启动swagger,true开启,fasle关闭;

Docket.select():设置哪些接口暴露给Swagger展示

Docket.select().api(selector):配置扫描接口的方式

basePackage(package):指定要扫描的包

any():扫描全部

none():不扫描

withClassAnnotation(annotation):扫描类上的注解，多数是一个注解的反射对象

withMethodAnnotation(annotation):扫描方法上的注解

@EnableOpenApi

@Configuration

**public** **class** SwaggerConfig {

**public** Docket docket() {

**return** **new** Docket(DocumentationType.***OAS\_30***)

.enable(**true**)

.apiInfo(apiInfo())

.select()

.apis(RequestHandlerSelectors.*withClassAnnotation*(Api.**class**))

.paths(PathSelectors.*any*())

.build();

}

**private** ApiInfo apiInfo() {

**return** **new** ApiInfoBuilder()

.title("爱旅行API文档")

.description("API文档描述,swagger-ui/index.html")

.contact(**new** Contact("wangtao", **null**, "wwangtaoc11@163.com"))

.version("springboot Version:"+SpringBootVersion.*getVersion*())

.build();

}

}

## 3.Swagger注解的使用说明

|  |
| --- |
| **@Api**：用在请求的类上，表示对类的说明  tags="说明该类的作用，可以在UI界面上看到的注解"  value="该参数没什么意义，在UI界面上也看到，所以不需要配置" |
| **@ApiOperation**：用在请求的方法上，说明方法的用途、作用  value="说明方法的用途、作用"  notes="方法的备注说明" |
| **@ApiImplicitParams**：用在请求的方法上，表示一组参数说明  **@ApiImplicitParam**：用在@ApiImplicitParams注解中，指定一个请求参数的各个方面  name：参数名  value：参数的汉字说明、解释  required：参数是否必须传  paramType：参数放在哪个地方  · header --> 请求参数的获取：@RequestHeader  · query --> 请求参数的获取：@RequestParam  · path（用于restful接口）--> 请求参数的获取： |
| @PathVariable  · div（不常用）  · form（不常用）  dataType：参数类型，默认String，其它值dataType="Integer"  defaultValue：参数的默认值 |
| **@ApiResponses**：用在请求的方法上，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  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**：用于响应类上，表示一个返回响应数据的信息  （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**：用在属性上，描述响应类的属性 |