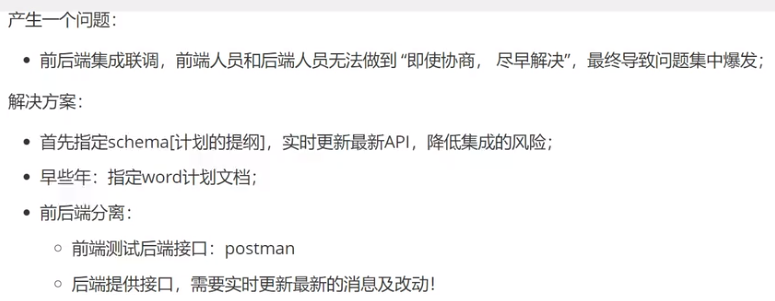
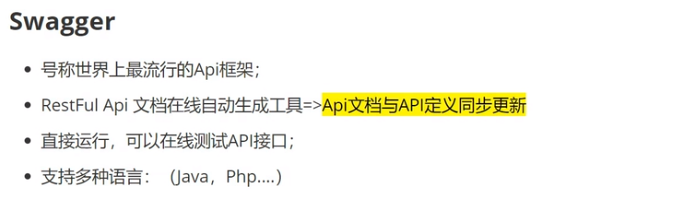
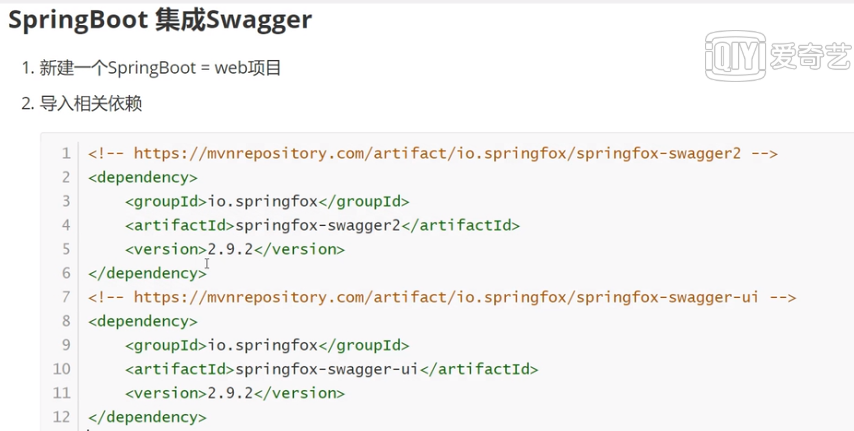
# Swagger简介







# Swagger集成Spring Boot



<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version>2.9.2</version>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

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

<version>2.9.2</version>

</dependency>

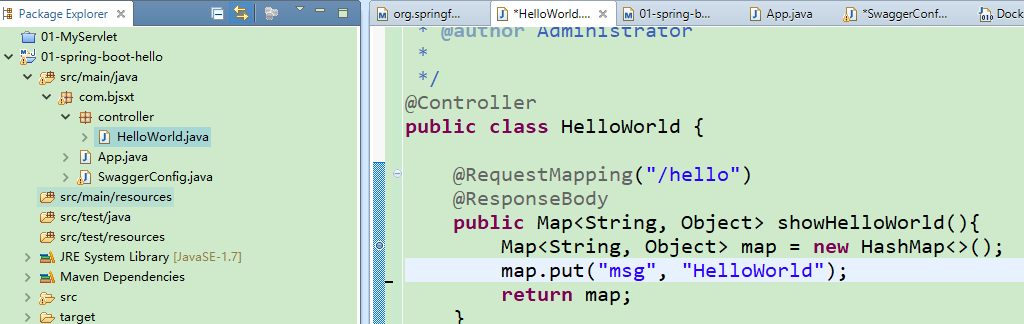
<dependency>

<groupId>com.google.guava</groupId>

<artifactId>guava</artifactId>

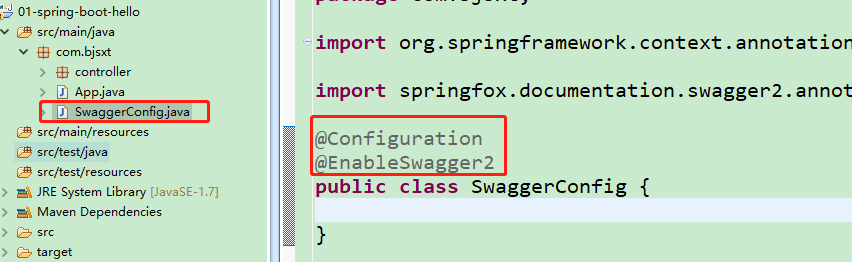
<version>26.0-jre</version>

</dependency>



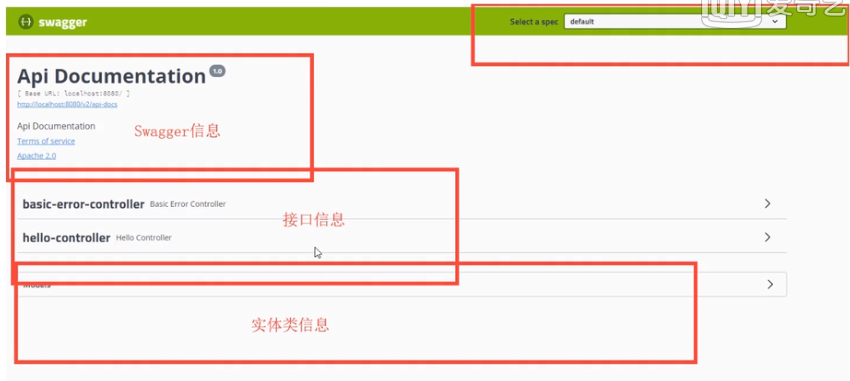
# 编写SwaggerConfig类

## 3.1.类注解

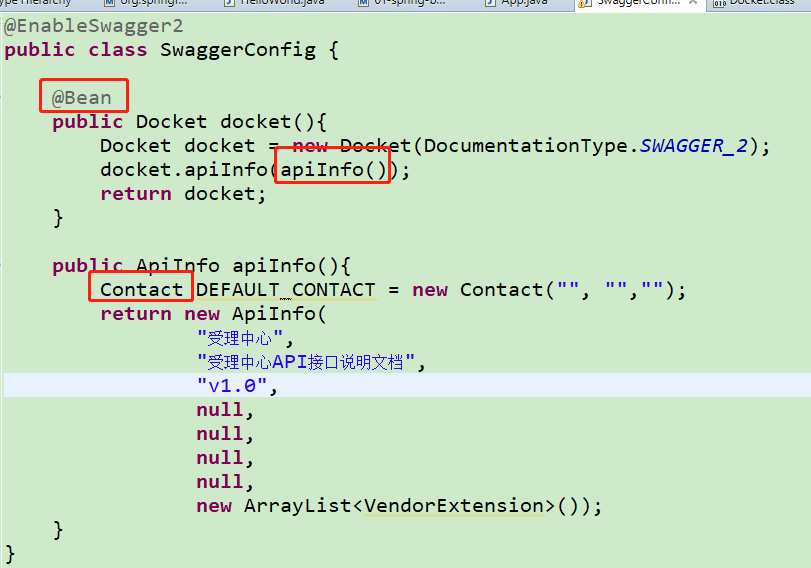


测试运行

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

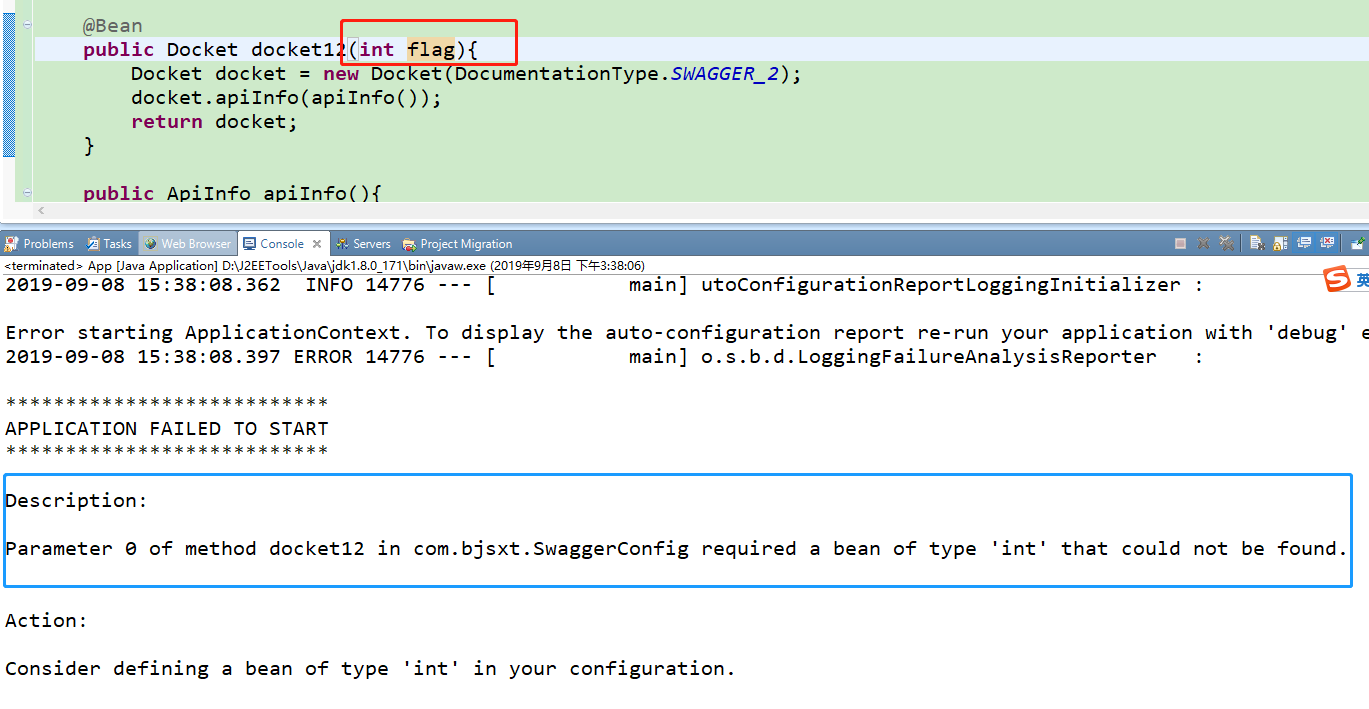


## 3.2.方法详解



* 最重要的是@Bean以及下面的方法，方法名自己随便定义啊，但是返回体一定要是Docket。如下是我自己做的试验，看看就明白啦~~~

正确的写法是不应该有入参或者将入参设置为规定的入参，但是，我自定义了一个入参，启动就报错啦。



* ApiInfo看源码就知道是啥意思啦。
* Contact看源码就知道啥意思啦。



## 3.3.配置接口扫描

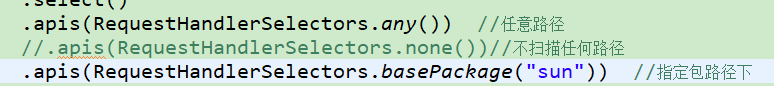


如果既有any ，又有none，那么最后是none。

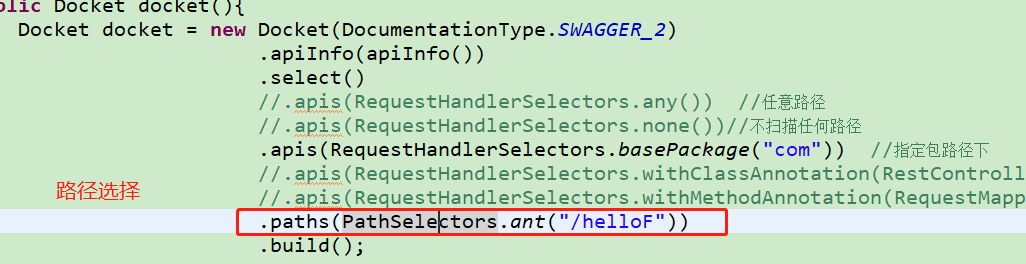


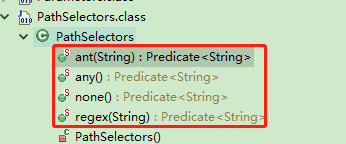


如果既有any，又有不存在的包，那么最后还是扫不到。

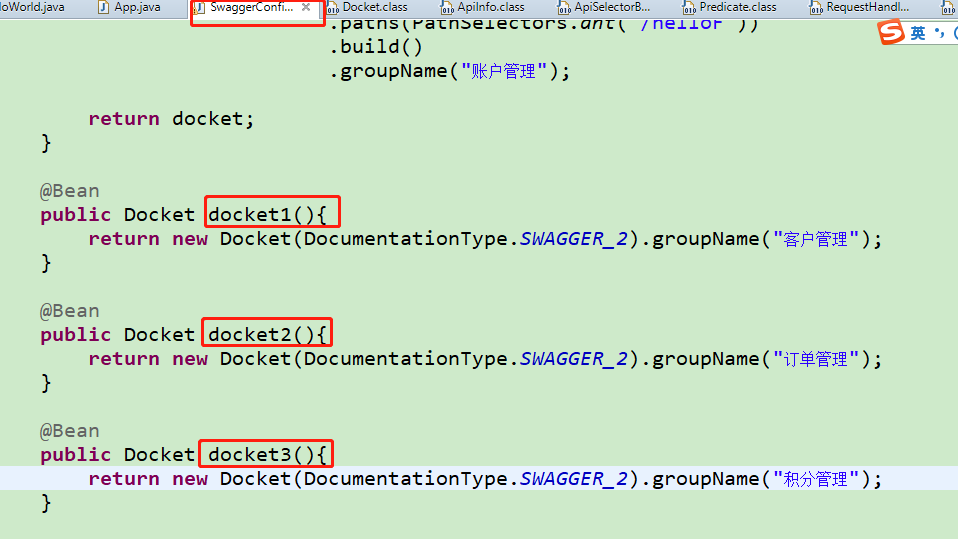


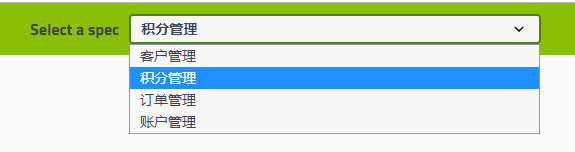




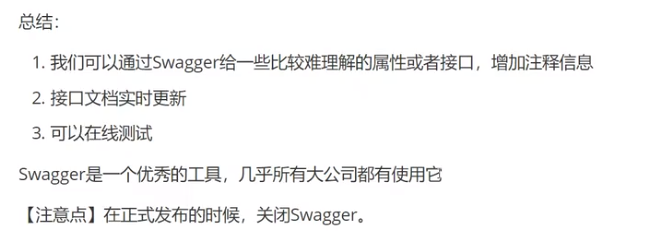


## 3.4.配置API分组

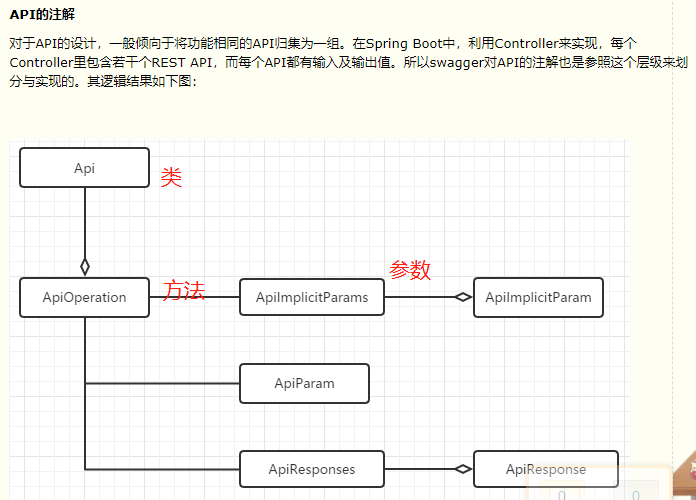




# 配置接口注解

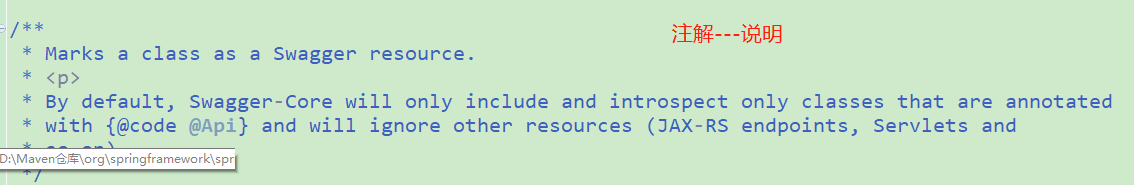




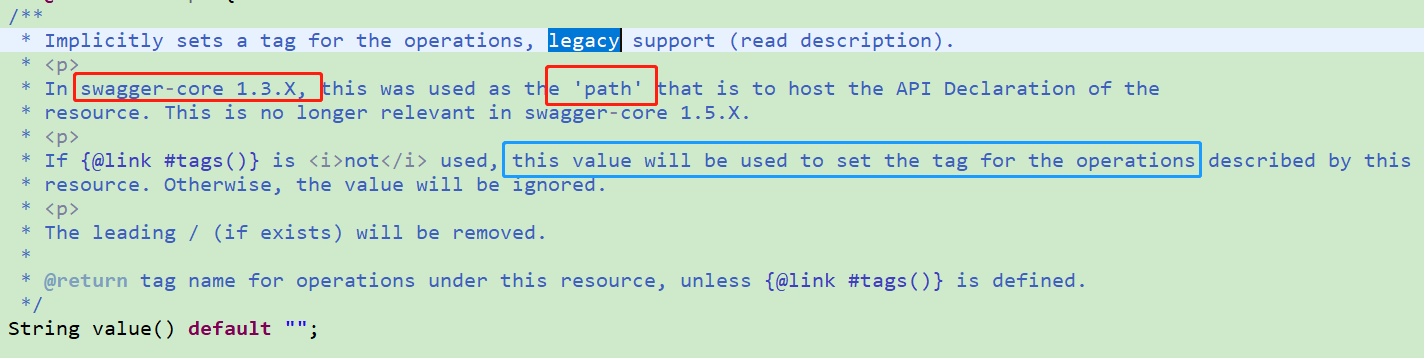


## 4.1.@Api注解-类

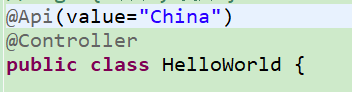


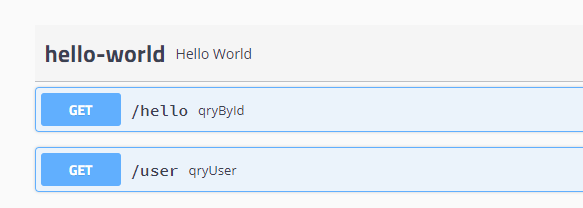


### 4.1.1.value

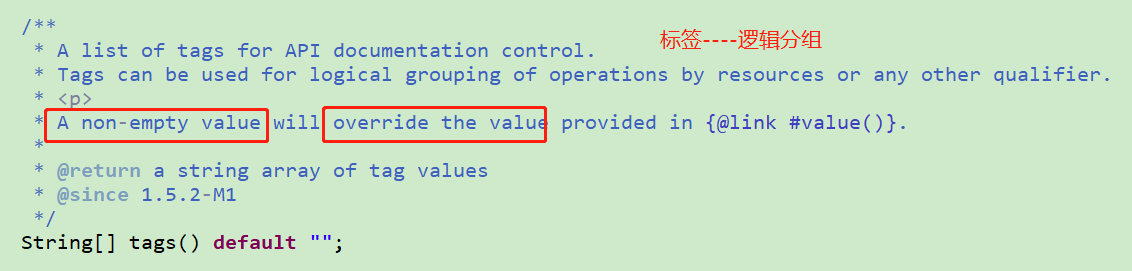


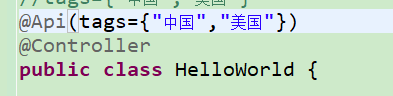
对于swagger2.9.2,我自己做了一下测试，发现value这个值根本不起作用啊。

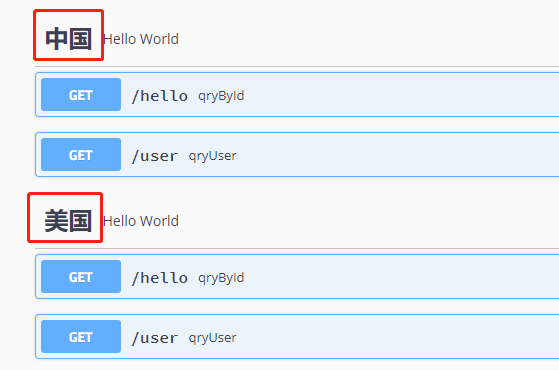




### 4.1.2.tags-类的分组/标签/概要

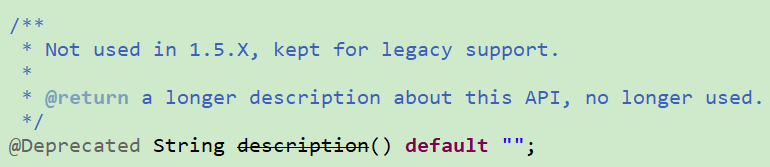




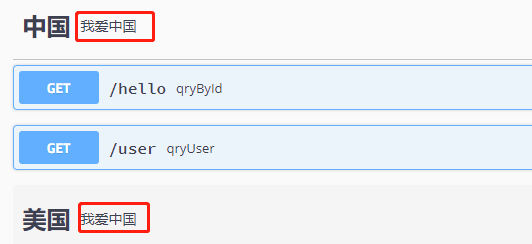


### 4.1.3.description-类的详细描述

被废弃了，但是依然还有用啊!~

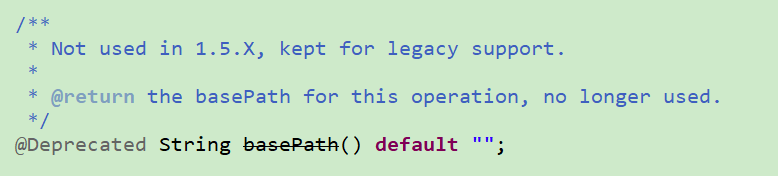




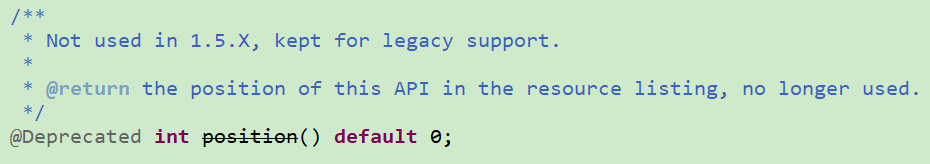


### 4.1.4.basePath

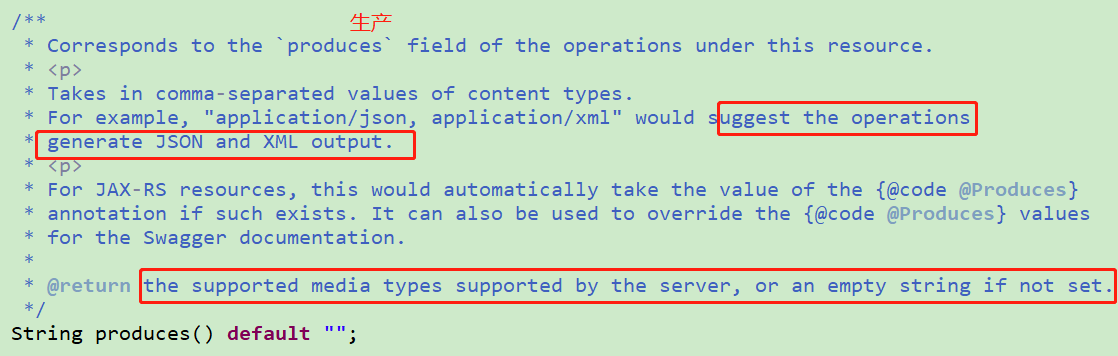
没用啦~~~



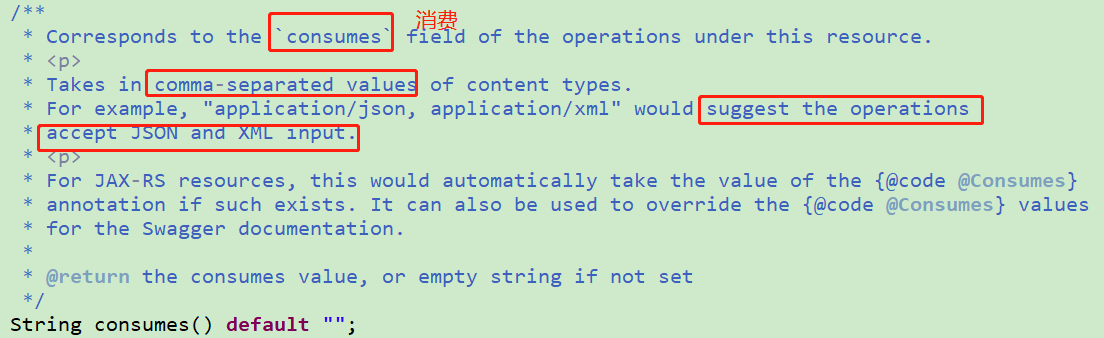
### 4.1.5.position



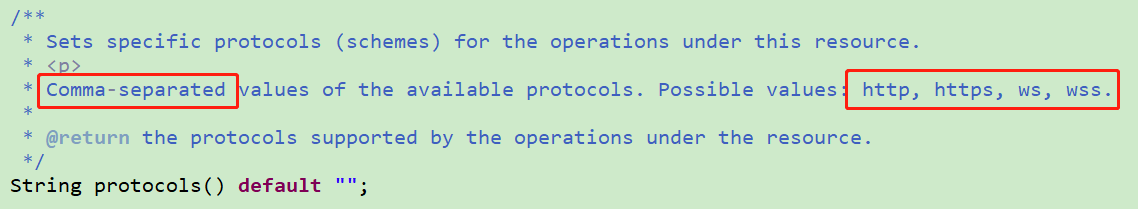
### 4.1.6.produces



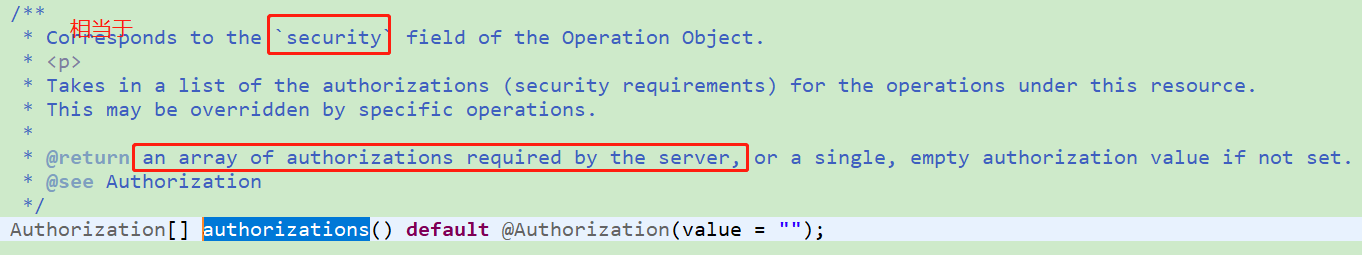
### 4.1.7.consumes



### 4.1.8.protocols

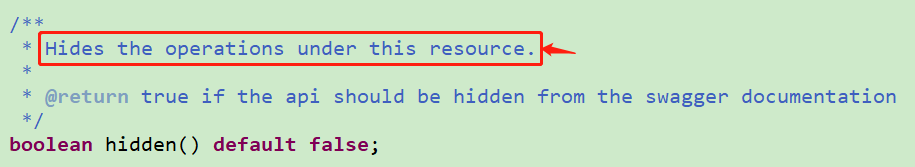


### 4.1.9. authorizations

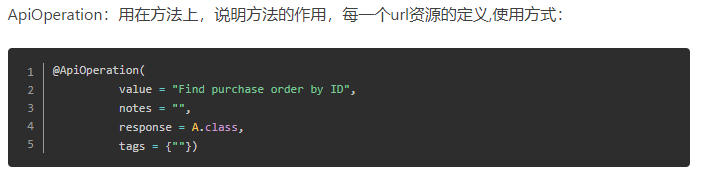


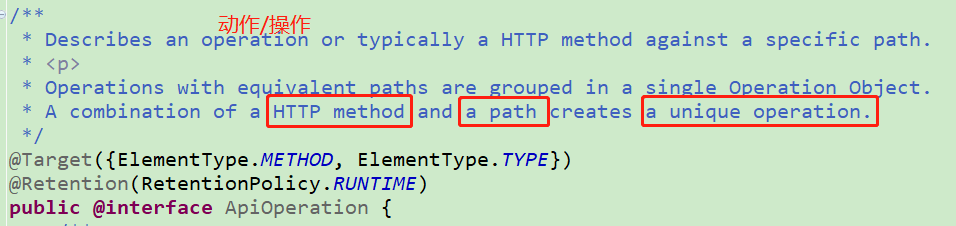
### 4.1.10. hidden

通过测试，发现不起作用

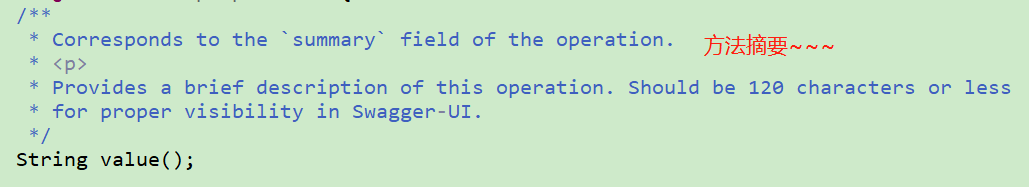


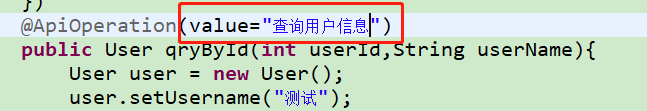
## [4.2.@ApiOperation](mailto:4.2.@ApiOperation)注解-方法





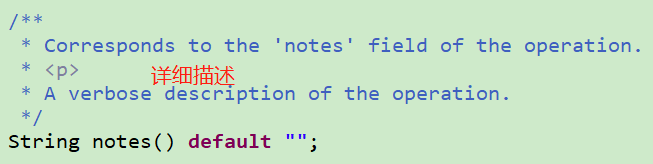
### 4.2.1.value-方法摘要

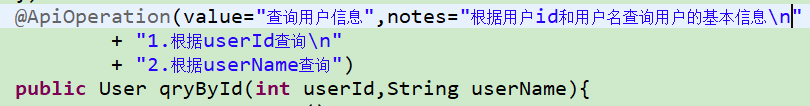


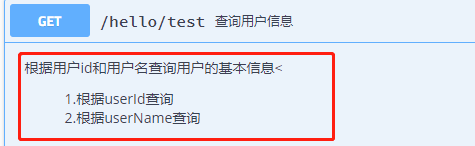




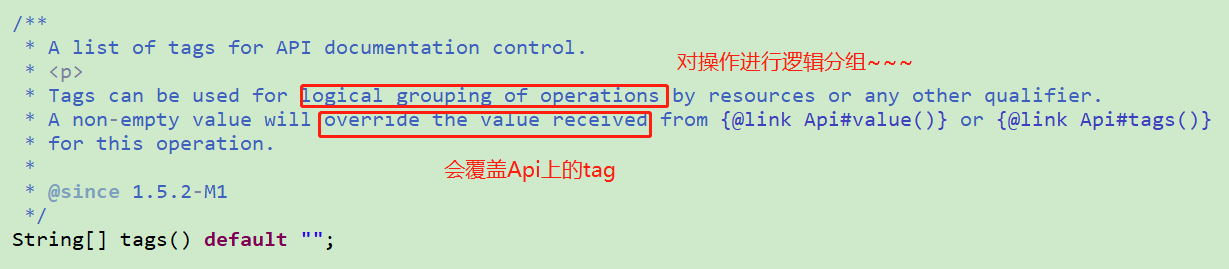
### 4.2.2.notes-方法详细描述

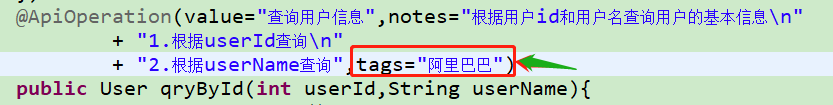






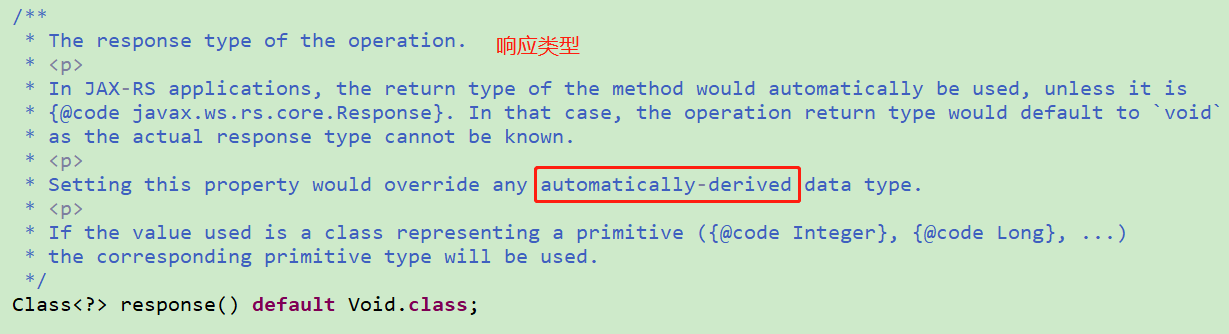
### 4.2.3.tags-方法的分组/标签

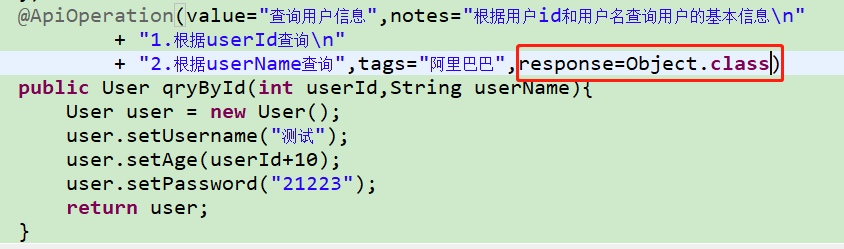


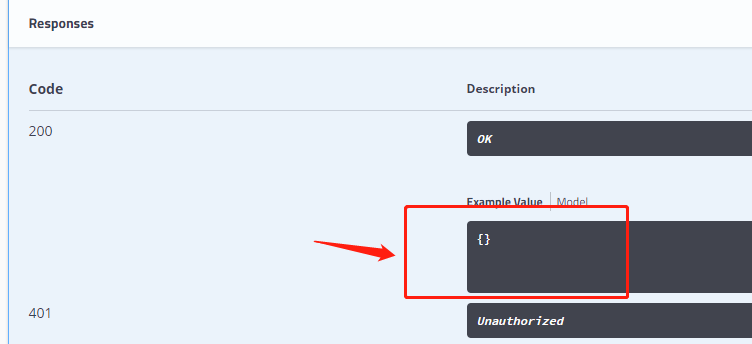


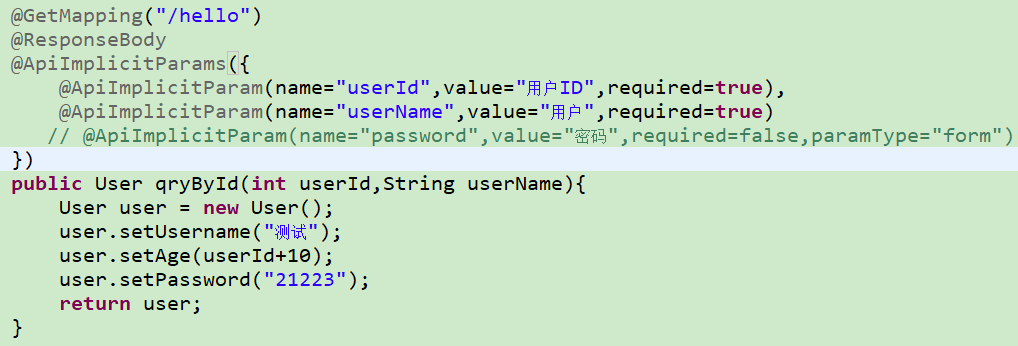


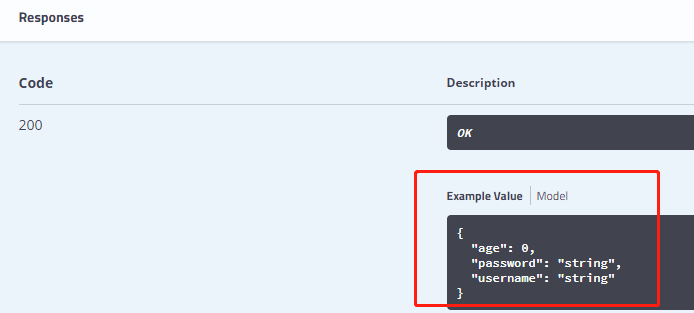
### 4.2.4.response



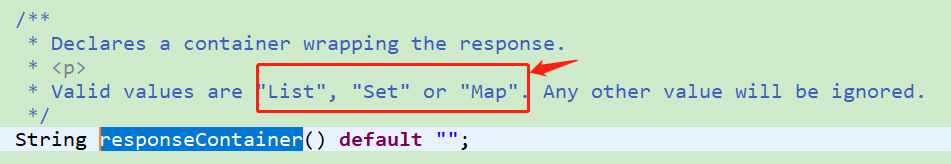




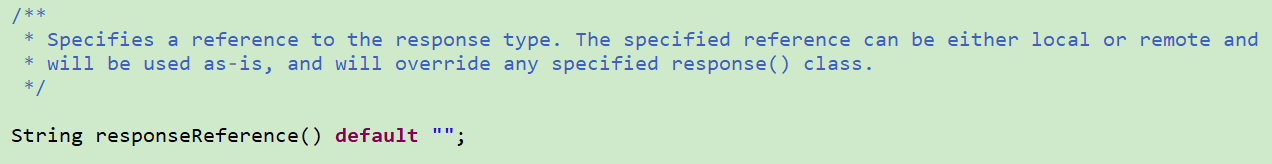




### 4.2.5. responseContainer

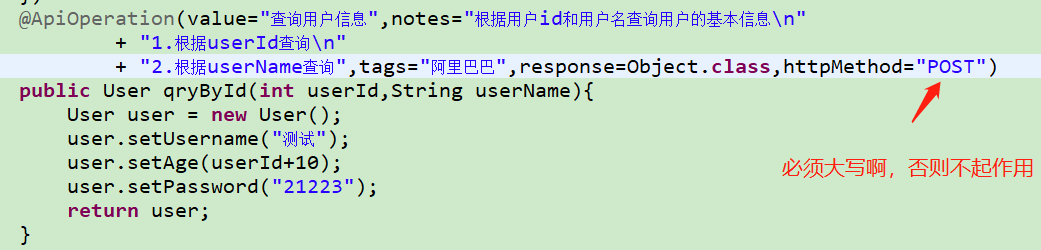
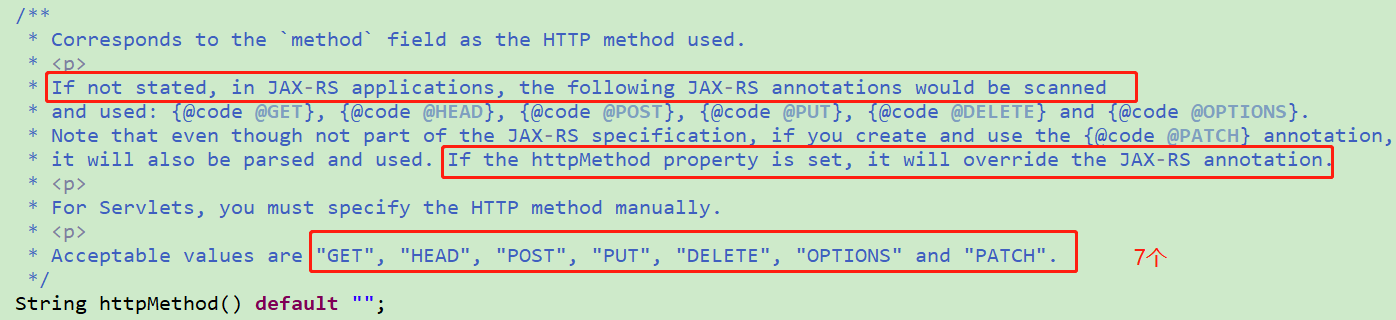


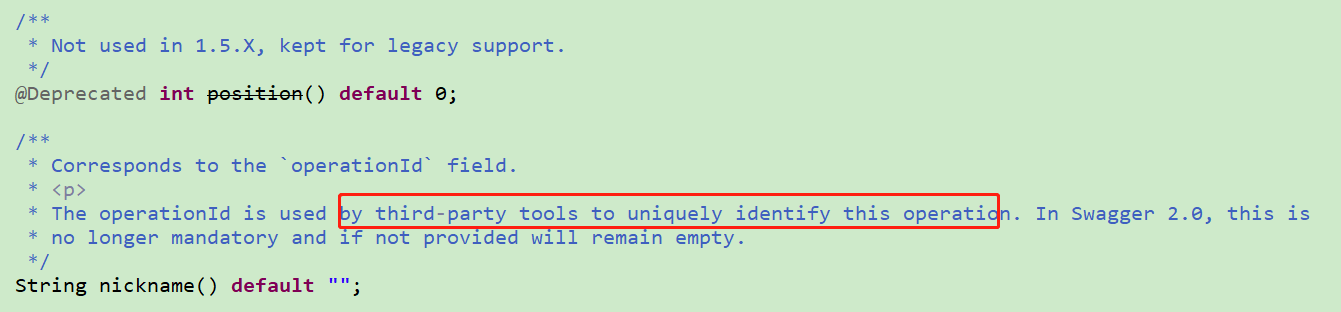
### 4.2.6.responseReference???



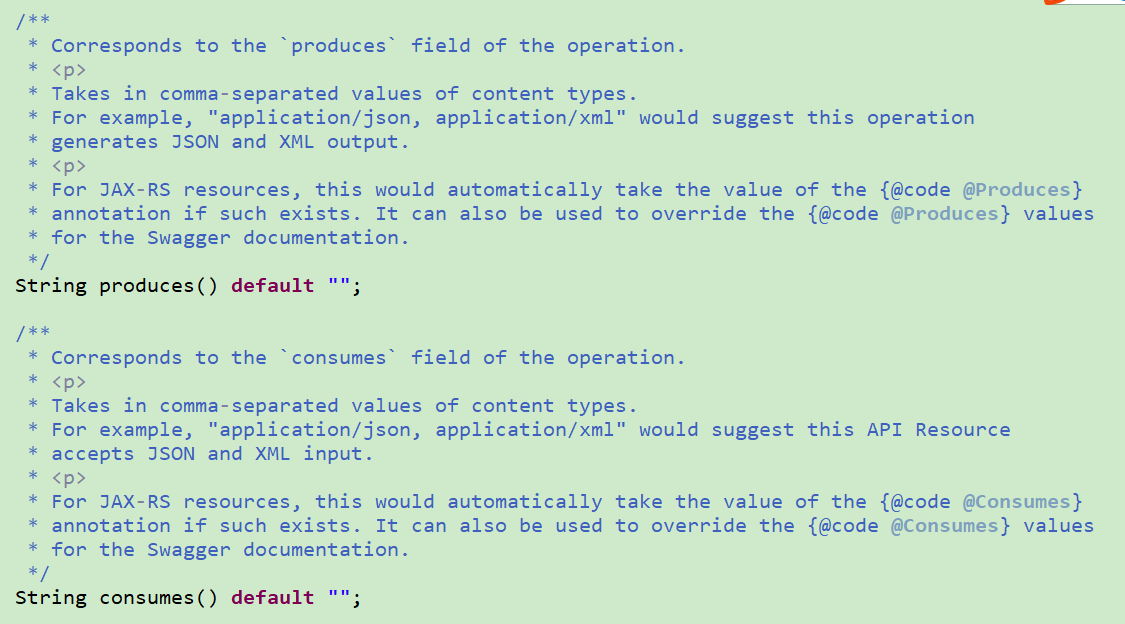
### 4.2.7.httpMethod

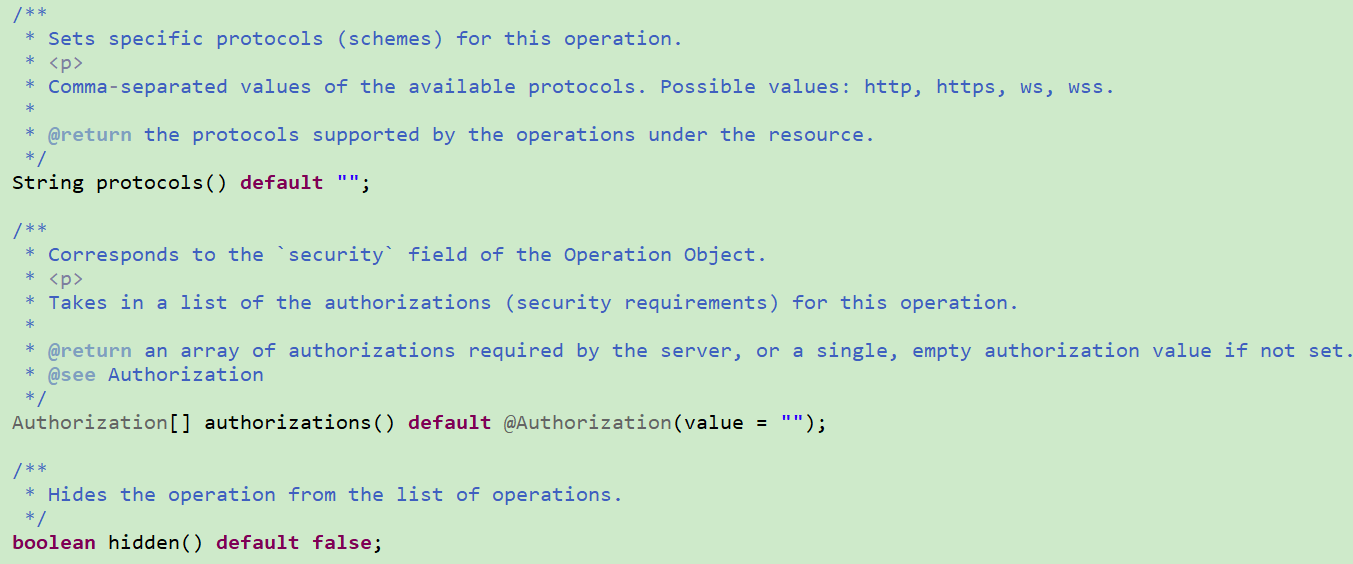
感觉没啥大用~~~

  
4.2.8.nickname

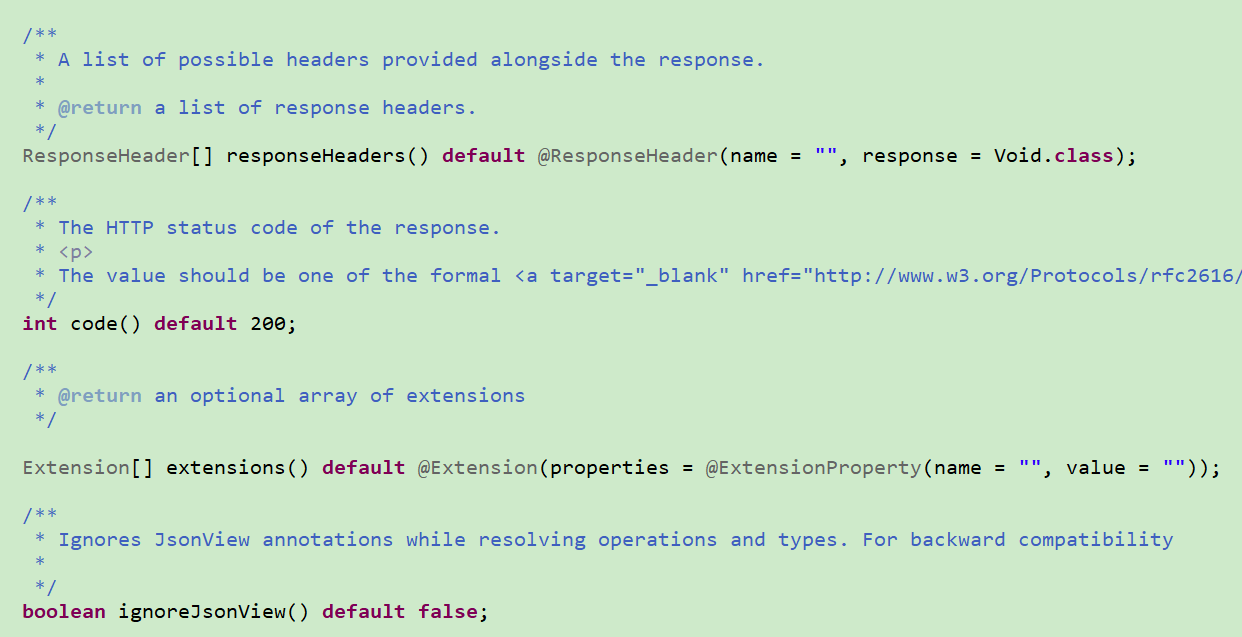


### 4.2.8.同Api

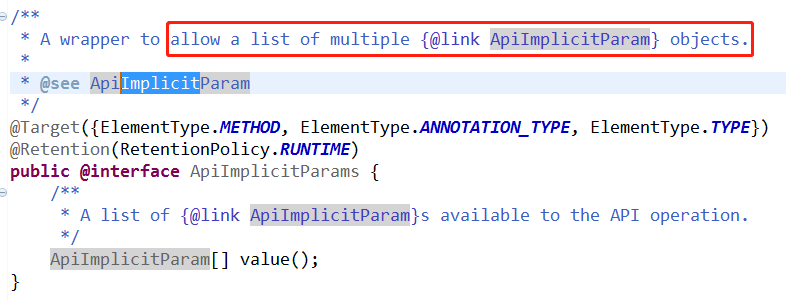




### 4.2.9.其他字段

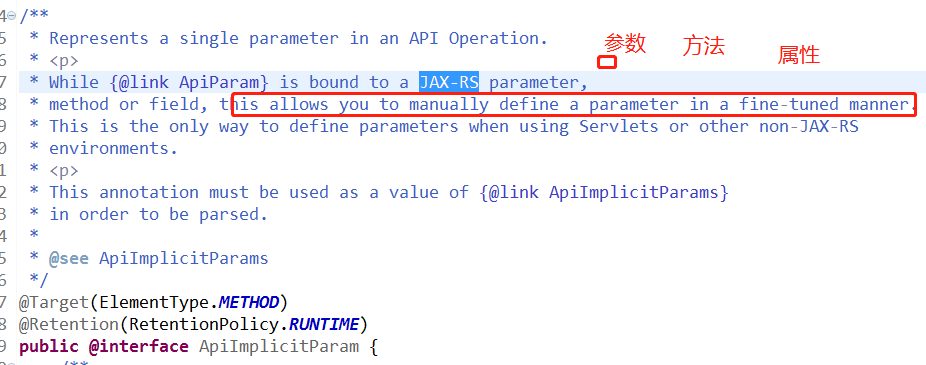


## [4.3.@ApiImplicitParams-](mailto:4.3.@ApiImplicitParams-)参数

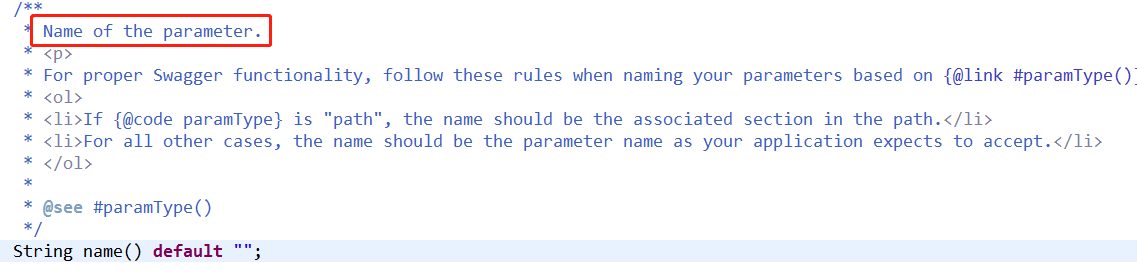


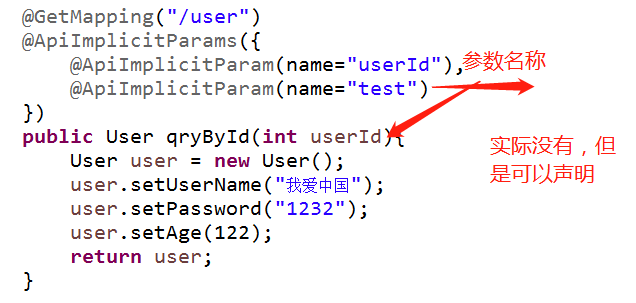
## [4.4.@ApiImplicitParam](mailto:4.4.@ApiImplicitParam)注解-参数

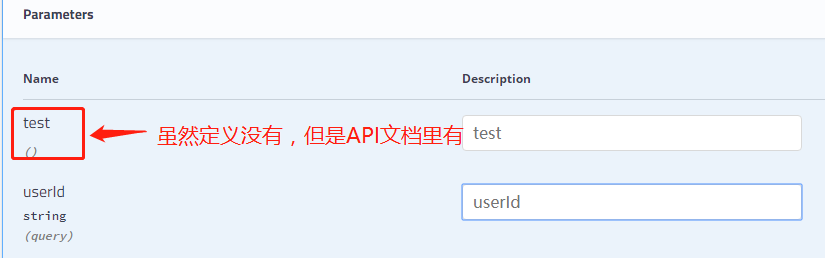


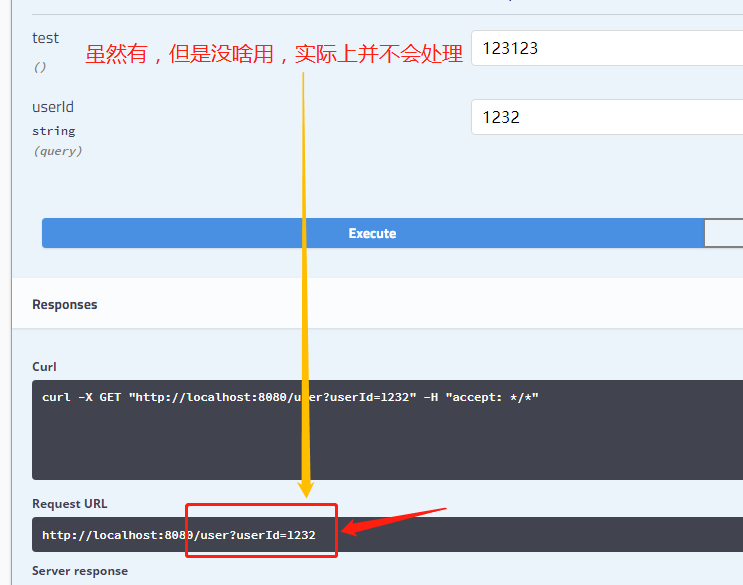


### 4.4.1.name-参数名称

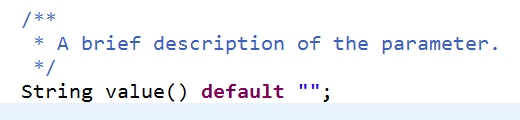




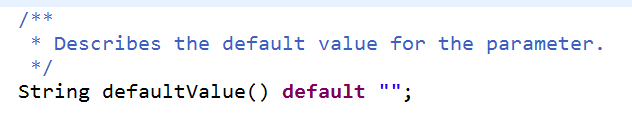


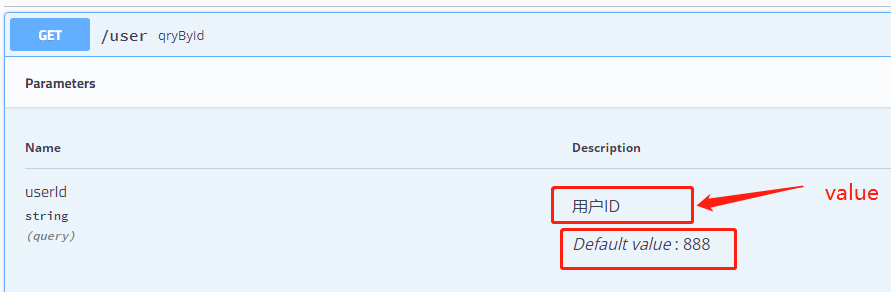


### 4.4.2.value-参数描述

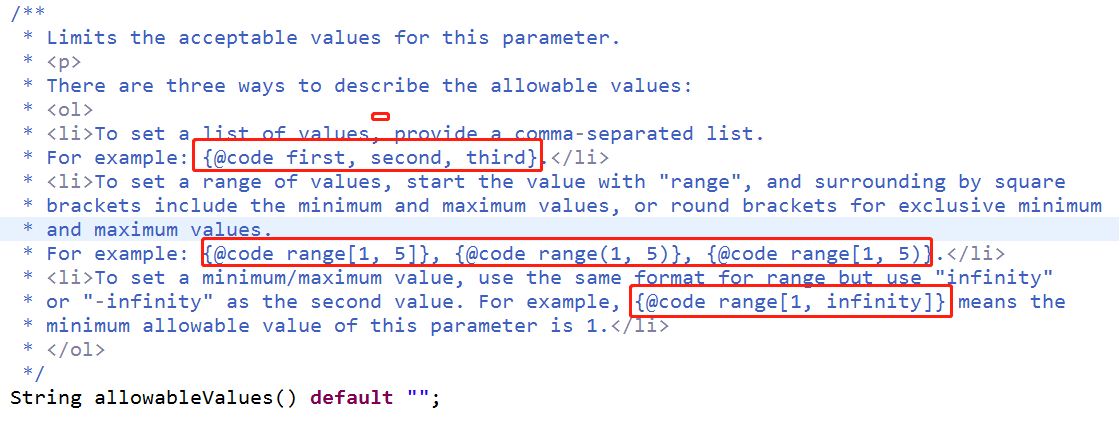


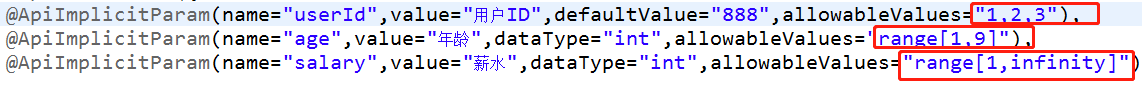
### 4.4.3.defaultValue-参数默认值





### 4.4.4.allowableValues-参数范围

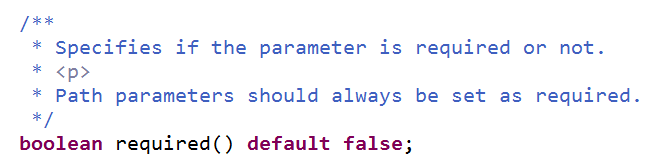


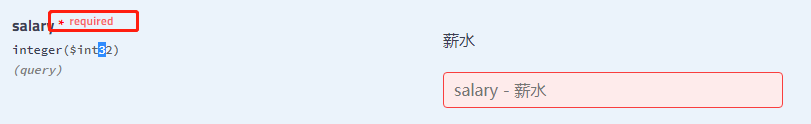




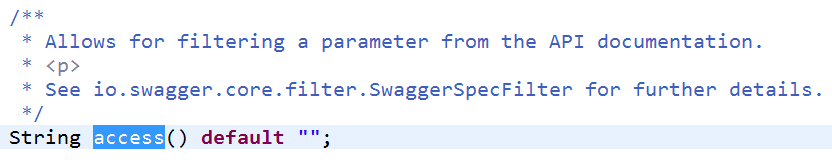


### 4.4.5.required-是否必填

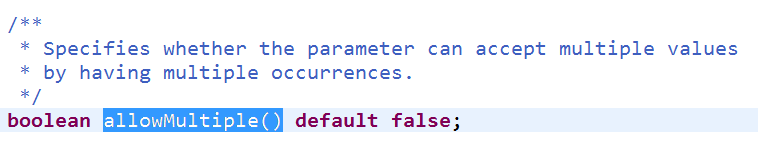




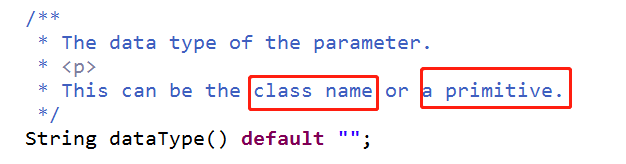
### 4.4.6.access-?



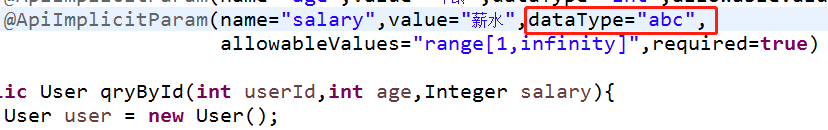
### 4.4.7. allowMultiple-?

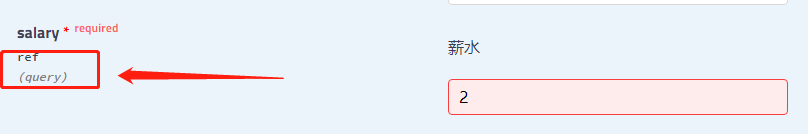


### 4.4.8.dataType-数据类型



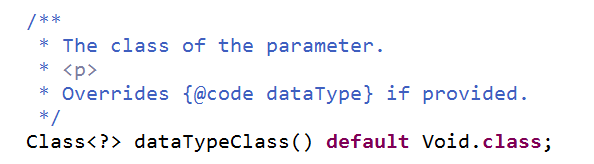
如果类名写错啦，页面上就变成ref啦，点击执行，执行不下去啊。



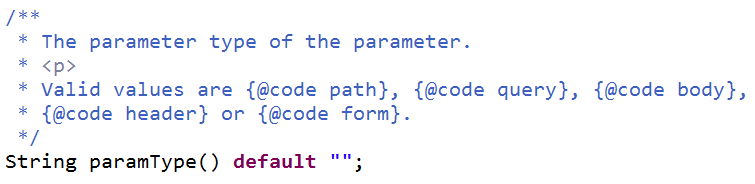


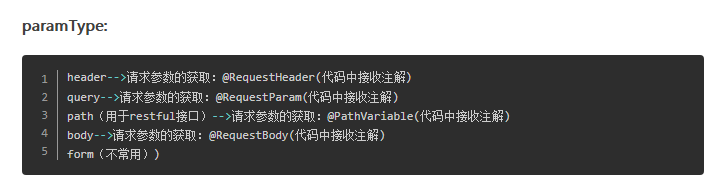
### 4.4.9.dataTypeClass-数据类型对应的类

和dataType的作用一致。



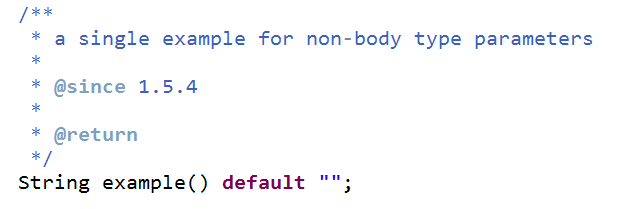
### 4.4.10.paramType-参数来源?



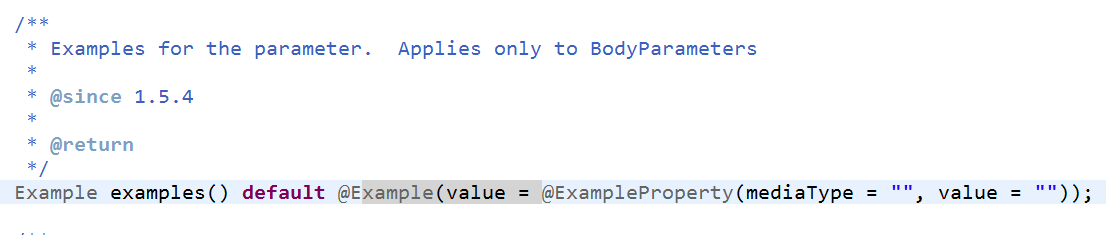


### 4.4.11.example-举例?

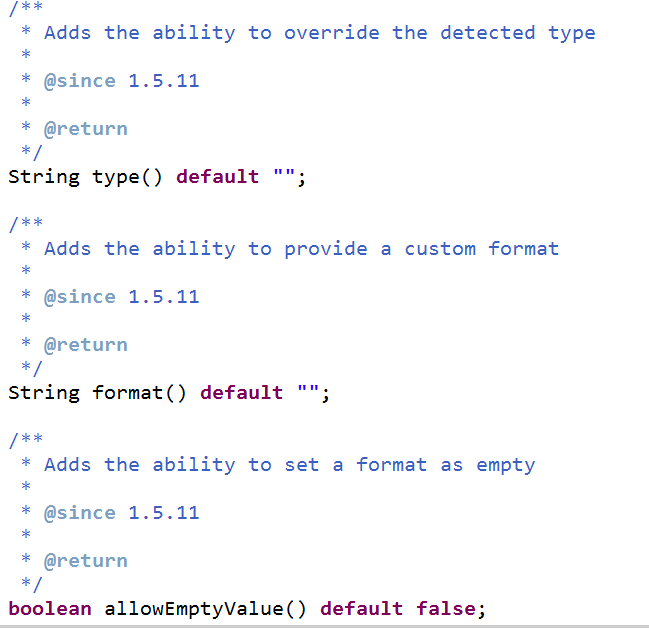
暂时不知道有啥用

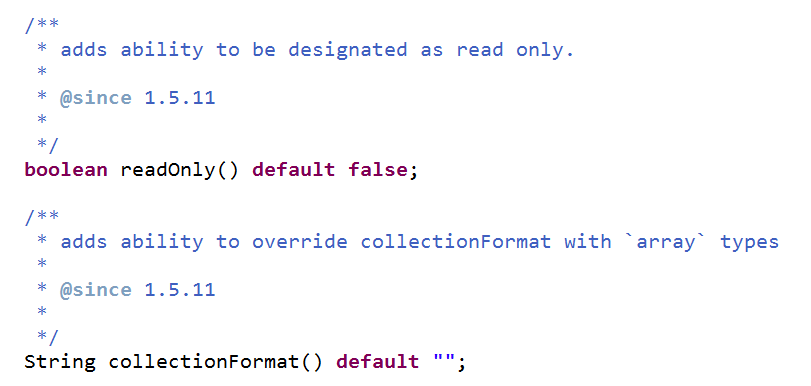


### 4.4.12.examples-?



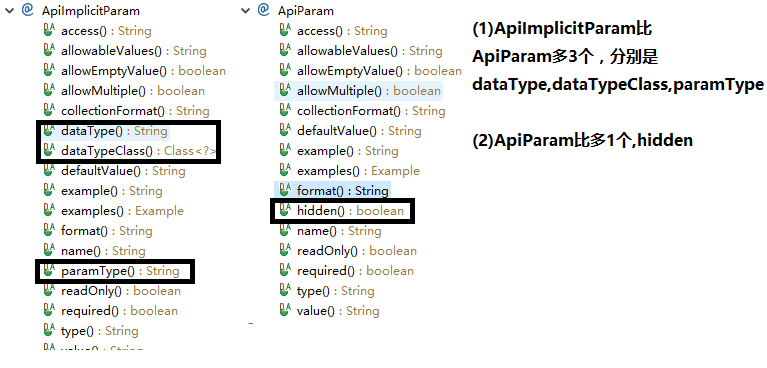
### 4.4.13.其它

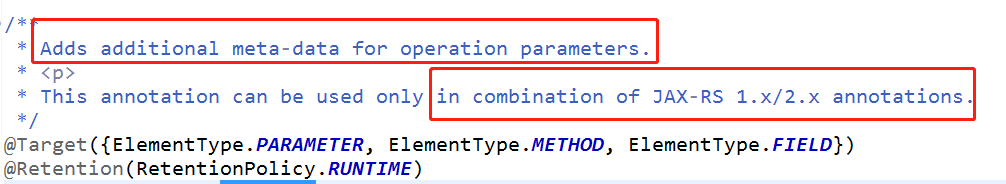


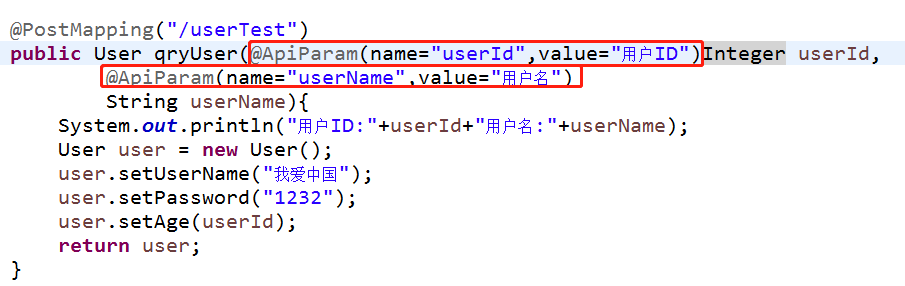


## [4.5@ApiParam](mailto:4.5@ApiParam)注解-参数

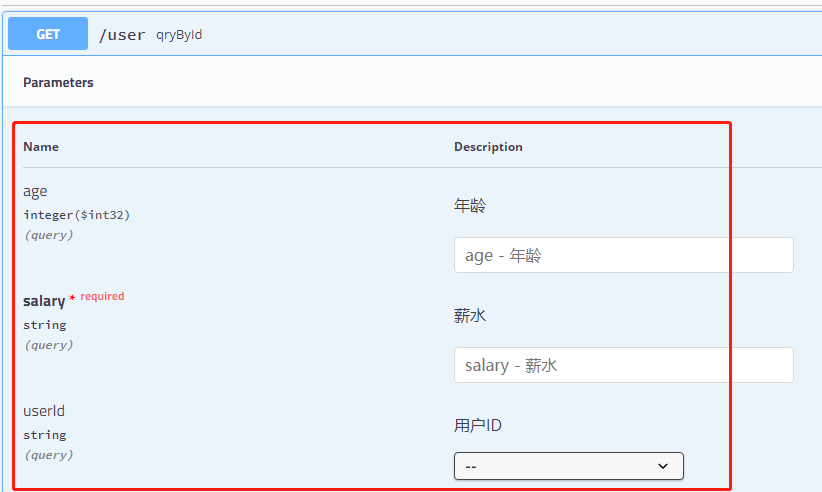
它和ApiImplicitParam有啥区别呢？





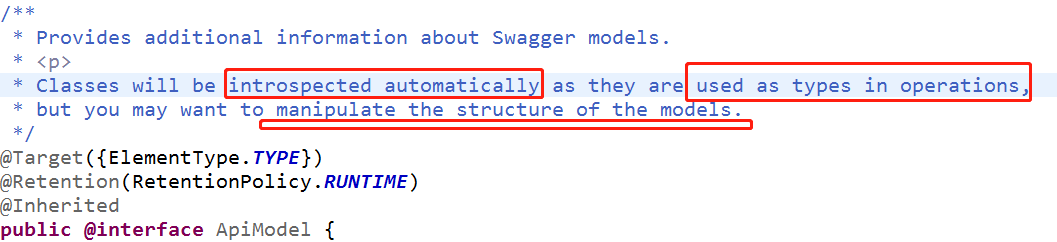


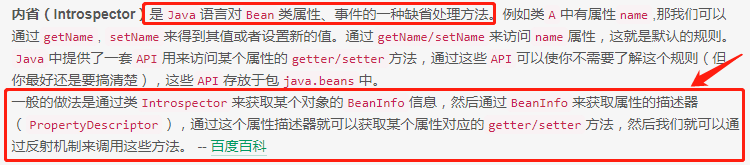




## [4.6.@ApiModel](mailto:4.6.@ApiModel)注解

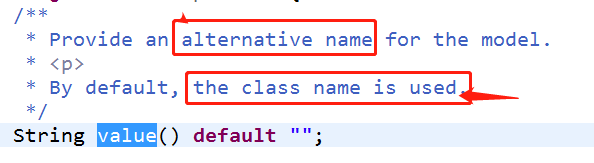


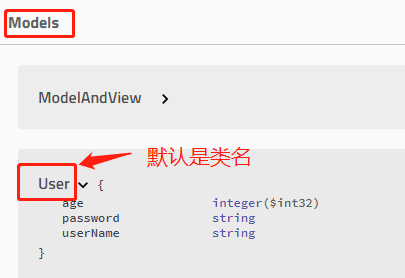




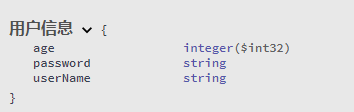
<https://www.cnblogs.com/peida/archive/2013/06/03/3090842.html>

### 4.6.1.value-Model名称

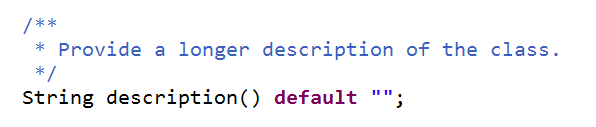


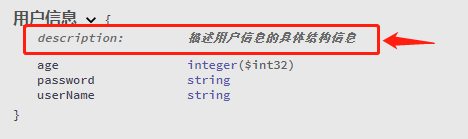




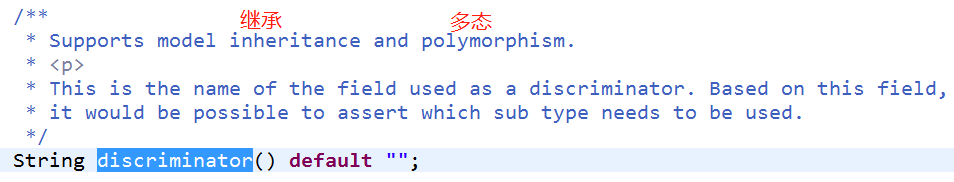


### 4.6.2.description-详细描述

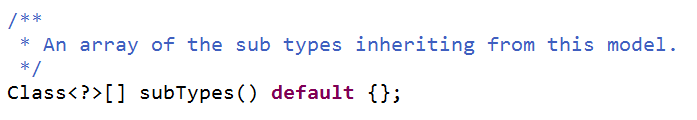




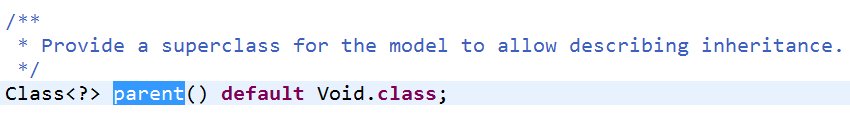
### 4.6.3. discriminator



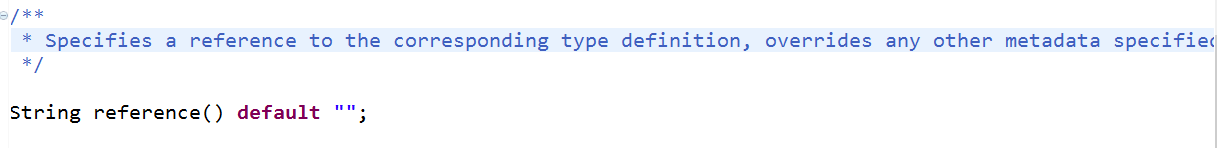
### 4.6.4.subTypes



### 4.6.5.parent



### 4.6.6.reference-引用?



## [4.7.@ApiModelProperty](mailto:4.7.@ApiModelProperty)注解

直接看官方源码吧，不在叙述啦。

