# 官网

https://swagger.io/

# 概念

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。

# 作用

1、接口文档在线自动生成。

2、功能测试

# springfox-swagger2 与 swagger 的区别

1、Swagger Spec 是一个规范。

2、Swagger Api 是 Swagger Spec 规范的一个实现，它支持 jax-rs, restlet, jersey 等等。

3、Springfox libraries 是 Swagger Spec 规范 的另一个实现，专注于 spring 生态系统。

4、Swagger.js and Swagger-ui 是 javascript 的客户端库，可以使用 Swagger Spec规范。

5、springfox-swagger-ui 仅仅是以一种方便的方式封装了 swagger-ui ，使得 Spring 服务可以提供服务。

总结：

1、Swagger 是一种规范。

2、springfox-swagger 是基于 Spring 生态系统的该规范的实现。

3、springfox-swagger-ui 是对 swagger-ui 的封装，使得其可以使用 Spring 的服务。

# 在maven中添加依赖

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

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

<version>2.2.2</version>

</dependency>

# 创建Swagger2的配置类



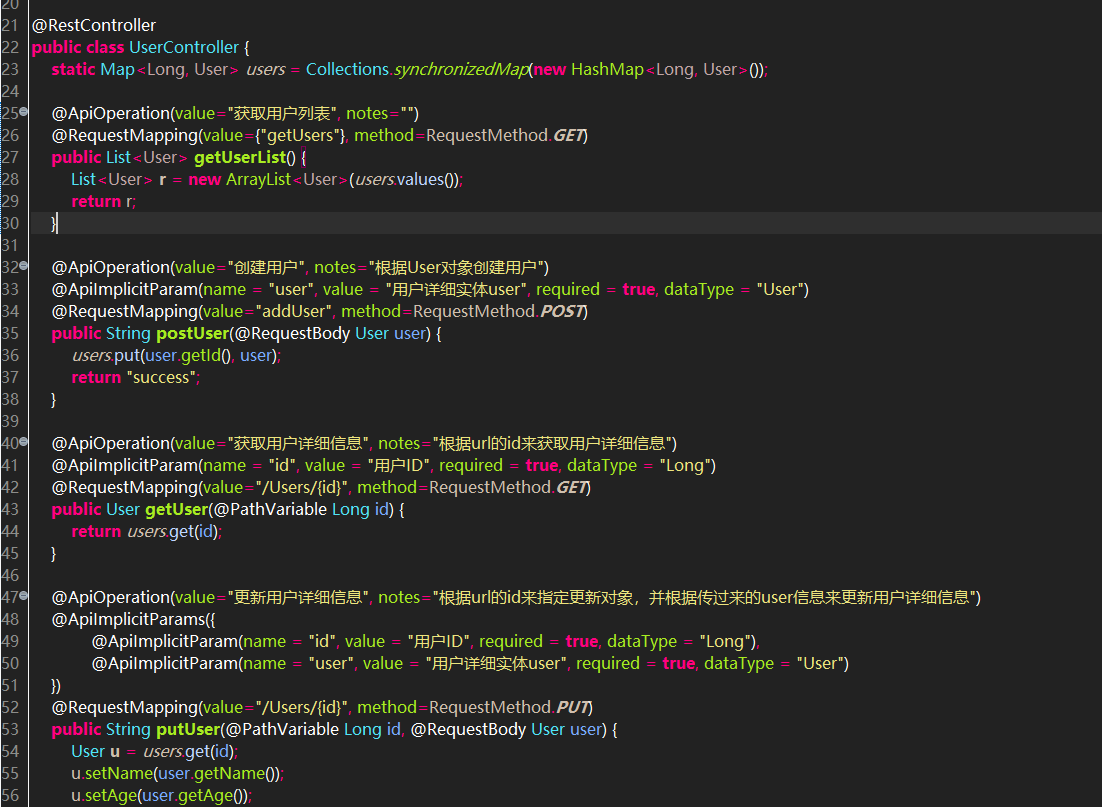
如上代码所示，通过@Configuration注解，让Spring来加载该类配置。再通过@EnableSwagger2注解来启用Swagger2。再通过createRestApi函数创建Docket的Bean之后，apiInfo()用来创建该Api的基本信息（这些基本信息会展现在文档页面中）。select()函数返回一个ApiSelectorBuilder实例用来控制哪些接口暴露给Swagger来展现，本例采用指定扫描的包路径来定义，Swagger会扫描该包下所有Controller定义的API，并产生文档内容（除了被@ApiIgnore指定的请求）。

# 编写接口类

## 基本操作

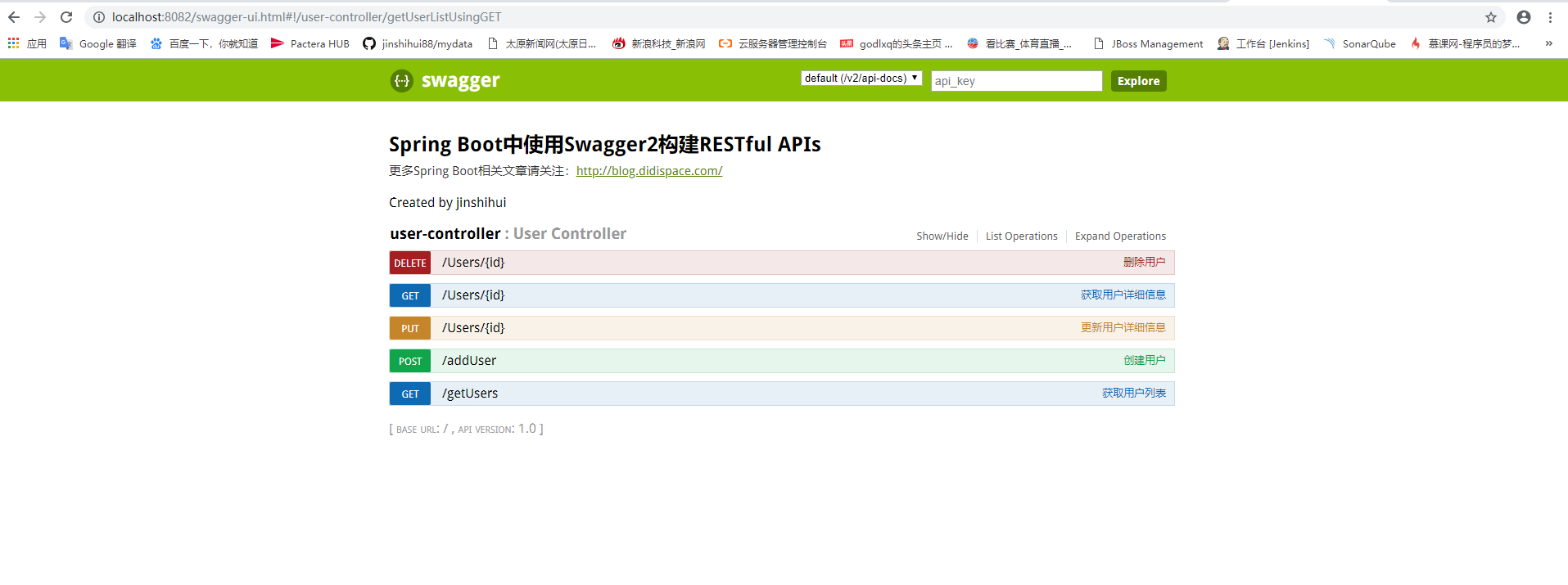
在完成了上述配置后，其实已经可以生产文档内容，但是这样的文档主要针对请求本身，而描述主要来源于函数等命名产生，对用户并不友好，我们通常需要自己增加一些说明来丰富文档内容。如下所示，我们通过@ApiOperation注解来给API增加说明、通过@ApiImplicitParams、@ApiImplicitParam注解来给参数增加说明。

代码如下：

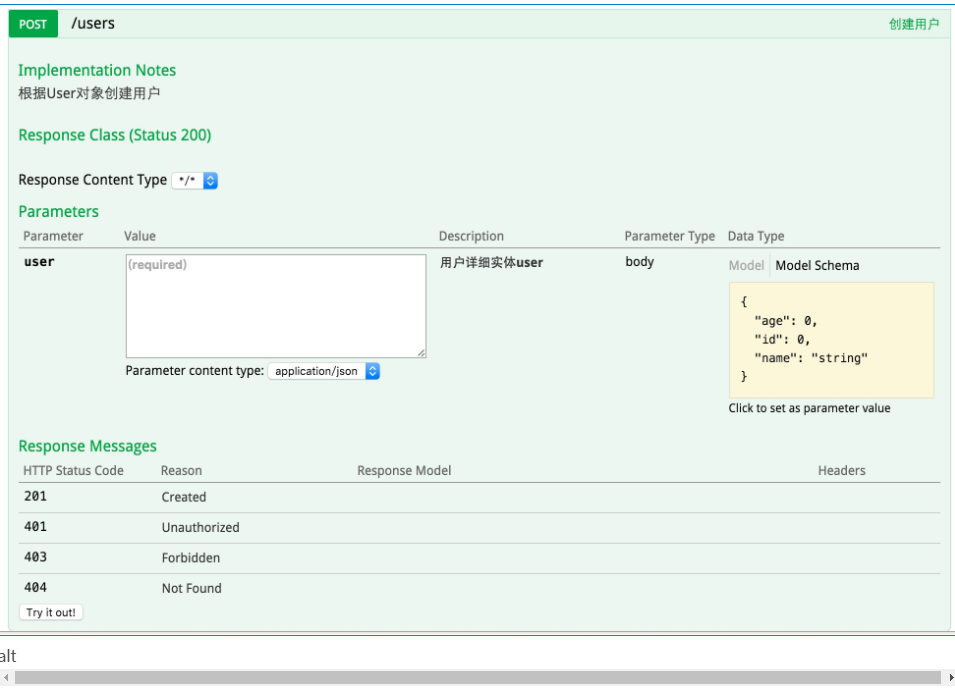


## 访问地址

完成上述代码添加上，启动Spring Boot程序，访问：<http://localhost:8080/swagger-ui.html>，如下图所示：



就能看到前文所展示的RESTful API的页面。我们可以再点开具体的API请求，以POST类型的/users请求为例，可找到上述代码中我们配置的Notes信息以及参数user的描述信息，如下图所示



## API文档访问与调试

在上图请求的页面中，我们看到user的Value是个输入框？是的，Swagger除了查看接口功能外，还提供了调试测试功能，我们可以点击上图中右侧的Model Schema（黄色区域：它指明了User的数据结构），此时Value中就有了user对象的模板，我们只需要稍适修改，点击下方“Try it out！”按钮，即可完成了一次请求调用。

相比为这些接口编写文档的工作，我们增加的配置内容是非常少而且精简的，对于原有代码的侵入也在忍受范围之内。因此，在构建RESTful API的同时，加入swagger来对API文档进行管理，是个不错的选择。

# Swagger2的注解使用

## @Api

用在类上，说明这个类的作用

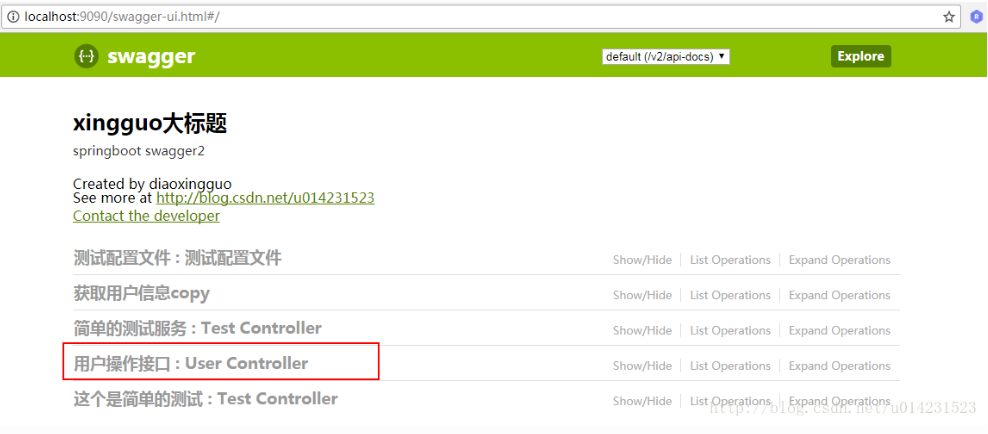
### 属性

value：也是说明，可以使用tags替代

tags：表示说明，但是tags如果有多个值，会生成多个list。

### 代码示例





## @ApiOperation

用在方法上，增加方法说明，表示一个http请求的操作。

### 属性

value：提供此操作的简要说明。为了在Swagger-UI中获得适当的可见性，应该是120个字符或更少。

notes：操作的详细描述。

## @ApiParam

用于方法，参数，字段说明，表示对参数添加元数据（说明或是否必填等）

### 属性

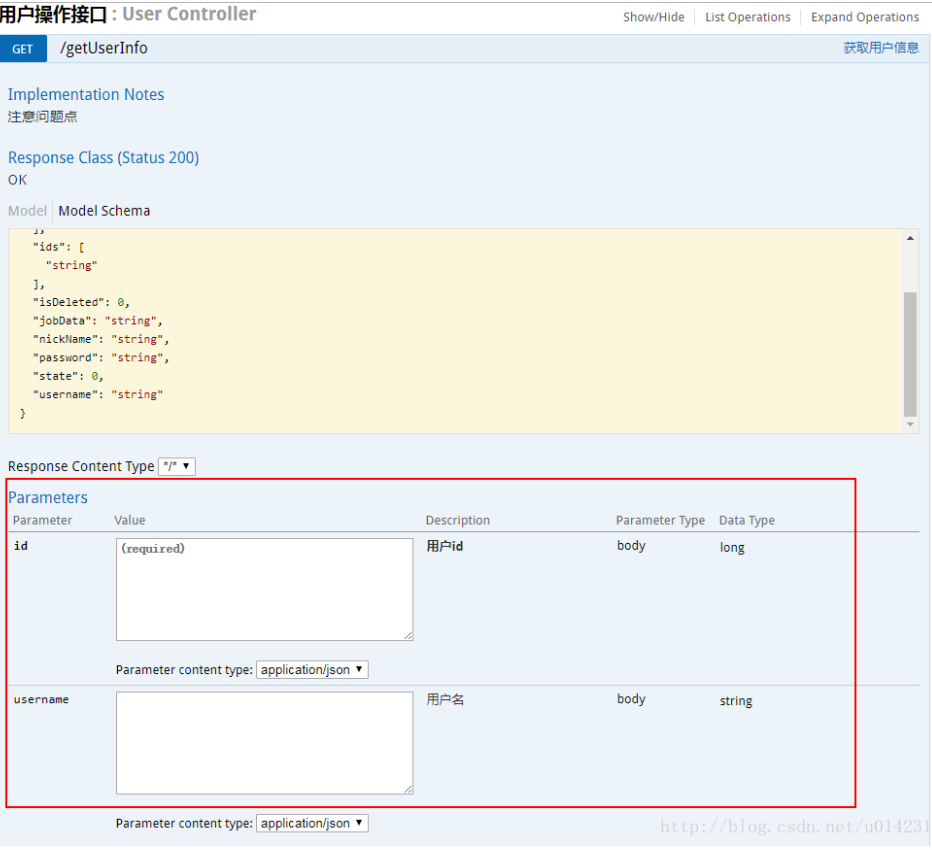
name：参数名

value：参数说明

required：是否必须

### 代码示例





## @ApiModel()

用在类上，表示对类进行说明，用于参数用实体类接收

## 属性

value：表示对象名

description：描述

## @ApiModelProperty()

用在方法、字段上，表示对model属性的说明或者数据操作更改

### 属性

value：字段说明

name：重写属性名称

dataType：重写属性类型

required–是否必填   
example–举例说明   
hidden–隐藏

### 代码示例





## @ApiImplicitParams

用在方法上，包含一组参数说明，包含多个 @ApiImplicitParam。

### 属性

name：参数名称

value：参数说明

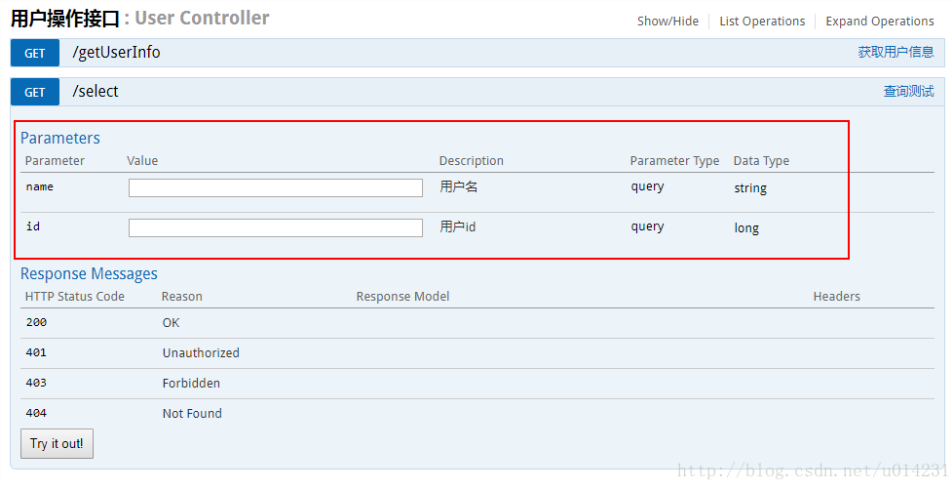
dataType：数据类型

paramType：参数类型

example：举例说明

### 代码示例





## @ApiImplicitParam

用在方法上，给方法入参增加说明

### 属性

name：要获得正确的Swagger功能，请在根据paramType命名参数时遵循以下规则：

（1）如果paramType是“path”，则名称应该是路径中的关联部分。

（2）对于所有其他情况，名称应该是应用程序期望接受的参数名称。

value：简洁的参数描述

required：指定是否需要参数，应始终根据需要设置路径参数。

dataType：参数的数据类型，这可以是类名或基元。

## @ApiResponses

用在方法上，用于表示一组响应

## @ApiIgnore

这个注解用在类、方法、参数上。表示这个方法或者类被忽略。

# 常用的类

## Docket

一个构建器，旨在成为swagger-springmvc框架的主要接口。为配置提供合理的默认设置和便捷方法。

### 方法

#### public Docket apiInfo(ApiInfo apiInfo)

设置包含在json ResourceListing响应中的api的元信息。

参数：

apiInfo 表示api的信息。

#### public ApiSelectorBuilder select()

启动api选择的构建器

返回：api选择构建器。 要完成构建api选择器，需要调用api选择器的构建方法，这将在调用构建方法时自动回退到构建docket。

#### public ApiSelectorBuilder apis(Predicate<RequestHandler> selector)

#### public ApiSelectorBuilder paths(Predicate<String> selector)

#### public Docket build()

## DocumentationType

文档类型

## PathSelectors

路径选择器

## ApiInfo

API的信息

## ApiInfoBuilder

API信息的构造器

### 方法

#### public ApiInfoBuilder title(String title)

更新api的标题

#### public ApiInfoBuilder description(String description)

更新api的描述

#### public ApiInfoBuilder termsOfServiceUrl(String termsOfServiceUrl)

更新服务条款地址

#### public ApiInfoBuilder contact(String contact)

更新负责此API的人员的联系信息

#### public ApiInfoBuilder version(String version)

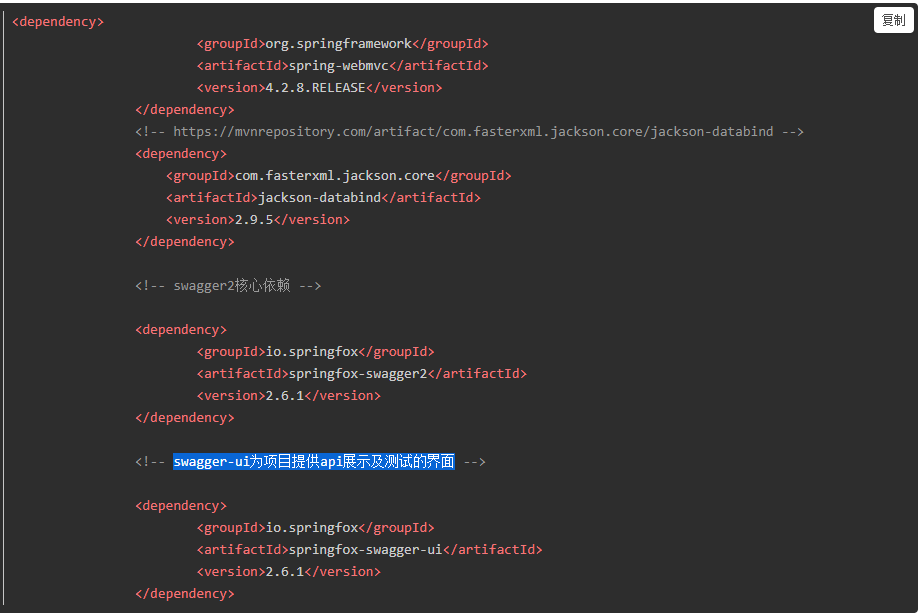
更新api的版本

#### public ApiInfo build()

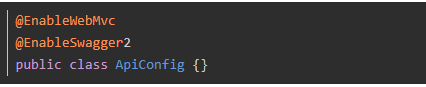
构建apiInfo对象

# SpringMVC集成springfox-swagger2构建restful API

## 1、配置pom.xml文件



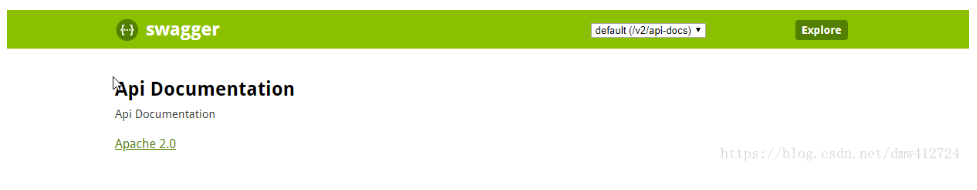
## 2、搭好个SpringMvc的框架，然后新建如下的类



这是一个空的类，然后把这个类做成bean，配置到Springmvc.xml文件里面去，启动起来后，这时候接口文档就可以直接使用了。

## 3、启动项目

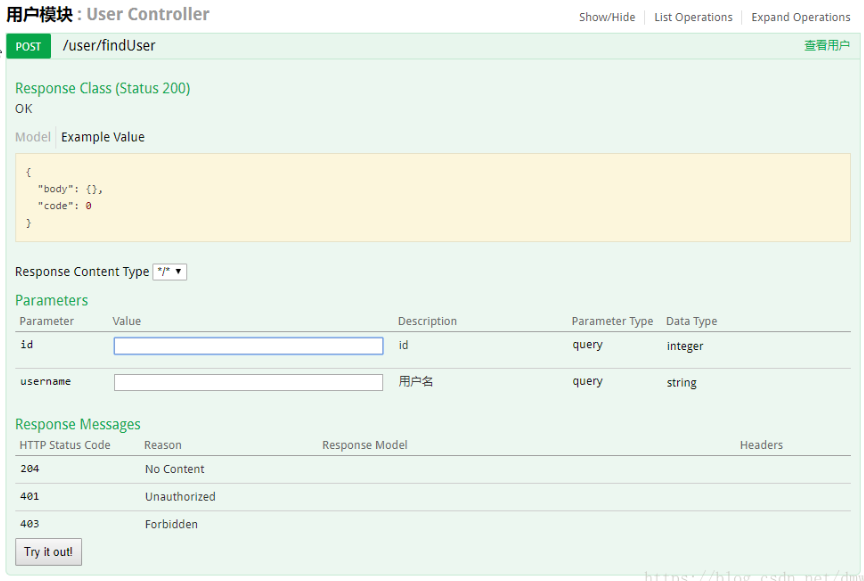
打开http://localhost:8080/${projectName}/swagger-ui.html.这里的projectName就是你的工程名



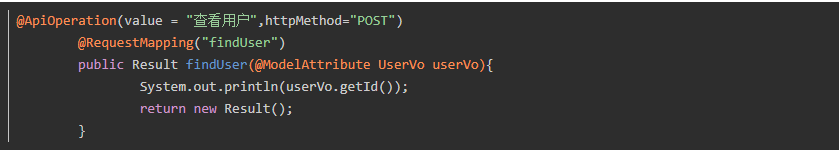
## 4、编写Controller



效果如下：



但是有时候这个id和username并不会被展开，而是一个Body的样子，那么这时候我们要在参数上加个注解就会自动解开。



那么如果这个参数对象我不想暴露怎么办？我只想接受其中的某个参数值，其他的并不想展露出来给API文档呢？那这时候就要忽略掉参数对象了，而是自行添加参数详情。



这里使用了@ApiIgnore的注解，忽略。这个注解不仅可以忽略参数，也可以加在方法上忽略掉url,让某些url不出现在API列表里面。

## 5、详细配置

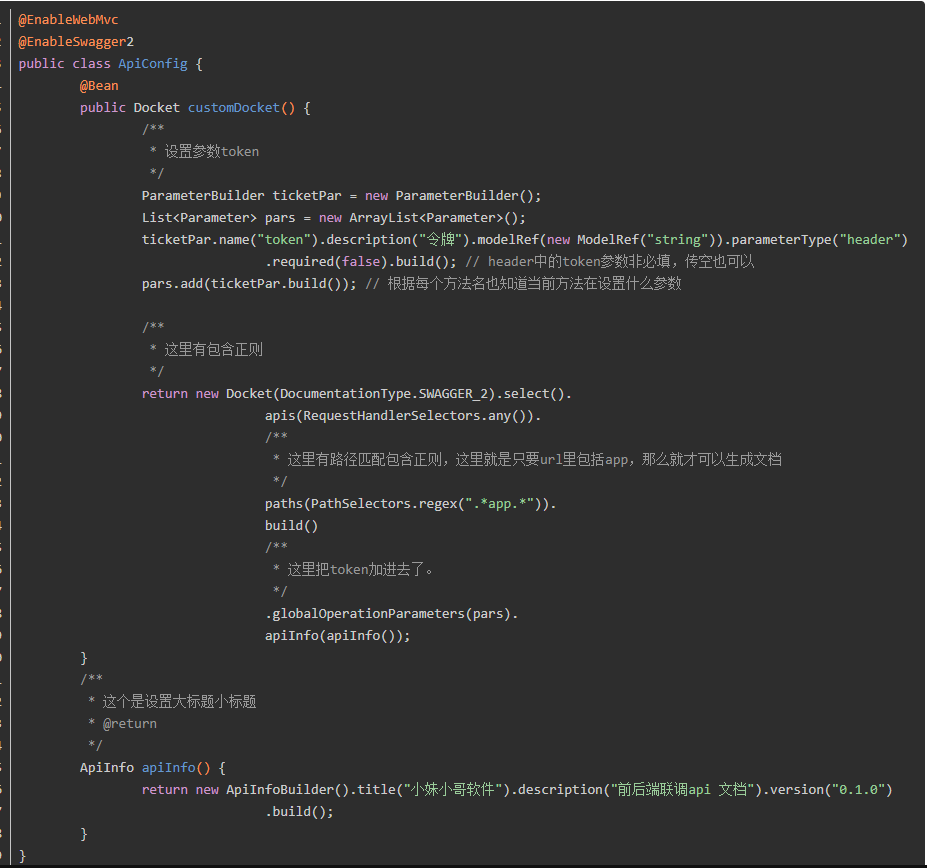
@ApiIgnore可以做到忽略，但是如果我们不想暴露的接口API比较多，那么给每个添加就不太合适了。所以这里我们要回到开始的地方，那个空的类里去配置一些东西。

以下配置包含3种东西。

（1）修改API页面的大小标题。

（2）修改API的包含路径正则，只针对正则里面的进行生成API。

（3）让每个访问都加入参数，我因为使用的是需要每个访问都要加入一个参数token。



注意的一些问题：

1、排序是按照@Api.value字典顺序进行排序的,但是tags会覆盖掉value。

2、接口列表无法单击展开，原因是@Api.tags使用了中文,如果想要中文说明描述,请使用那个@Api.discrption。