Spring MVC的注解

# @RequestMapping

## @RequestMapping功能

Spring MVC使用**@RequestMapping注解**为控制器**类或内部方法**指定可以处理哪些URL请求，也就是声明控制器的**handle方法**。**@RequestMapping** 是一个用来处理**请求地址映射**的注解，可用于类或方法上。

@RequestMapping 来映射请求地址，也就是通过它来指定控制器可以处理哪些URL请求，类似于struts的action请求。

## @RequestMapping修饰类和方法

注解**@RequestMapping**在**类级别和方法级别**层面都可标注，用来确定将被调用方法的URL。用于类上，表示类中的所有响应请求的方法都是以该地址作为**父路径**；用于方法上，表示**在类的父路径下追加方法上注解中的地址**将会访问到该方法。在类上可以使用也可以不使用@RequestMapping，但是在handle方法上必须使用@RequestMapping。

按照一般的开发习惯：类上的@RequestMapping一般就是利用**类名**作为**父路径，可以指定多层**，如@RequestMapping(value='/xxx/xxx/类名')；然后内部方法上利用**方法名**指定，一般就是一层路径，如@RequestMapping(value='/方法名')。

提供**初步的请求映射信息**，相当于WEB应用的根目录。

### 在方法处标注

提供进一步的细分映射信息，紧接着类定义处的URL。

在标有**@Controller**的类中，利用**@RequestMapping**可以将该类中的方法变成**handler**方法，用来处理请求。

## @RequestMapping的属性（4个）

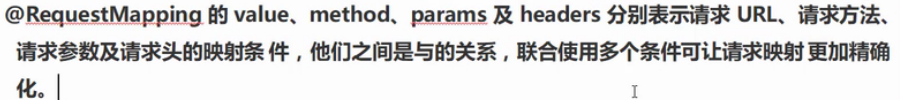
@RequestMapping具有四个属性：value、method、params、headers。

分别为映射请求路径（**value属性）**、请求方法（**method属性**）、请求参数（**parmas属性**）或请求头（**headers属性**）。多个属性同时指定的时候，利用逗号隔开，属性之间的关系是“与”的关系。

**@RequestMapping**一般情况必须指定属性value，用来指定对应的URL，这就是粗略映射。

当@RequestMapping**只指定一个属性value时**，此时value可省略不写，即@RequestMapping(value="/hello")等价于@RequestMapping("/hello")。

## @RequestMapping的精准化映射



### 映射路径value

指定方式：@RequestMapping(value="/hello")；当@RequestMapping**只指定一个属性value时**，此时value可省略不写。

### 映射请求方法method

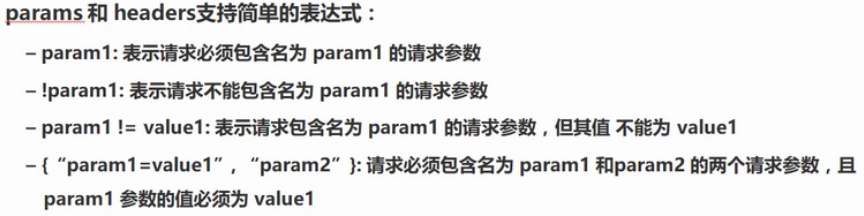
指定method需要一个枚举类**RequestMethod**。

指定方式：@RequestMapping(value="/hello",method=RequestMethod.POST)

详见后面的RequestMethod介绍。



### 映射请求参数params与映射请求头headers



**params**通过一个数组定义,如params={"xxx1=yyy1","xxx2=yyy2","xxx3","xxx4"}

意思是，必须有参数值为yyy1的xxx1参数，值为yyy2的xxx2参数，必须具有参数xxx3和xxx4参数，值可以为任意。

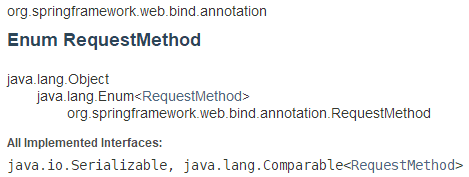
如：@RequestMapping(value="/hello",method=RequestMethod.POST,params={"user=zhaohong","passwd=zhao12","code"})

**headers**的使用方法与params相同，如headers={"Accept=text/html"}

# RequestMethod枚举类的介绍

## RequestMethod类：定义HTTP的请求方法。

public enum **RequestMethod** extends java.lang.Enum<RequestMethod>



Java 5 enumeration of HTTP request methods. Intended for use with the [RequestMapping.method()](https://docs.spring.io/spring/docs/5.0.5.BUILD-SNAPSHOT/javadoc-api/org/springframework/web/bind/annotation/RequestMapping.html#method--) attribute of the [RequestMapping](https://docs.spring.io/spring/docs/5.0.5.BUILD-SNAPSHOT/javadoc-api/org/springframework/web/bind/annotation/RequestMapping.html) annotation.

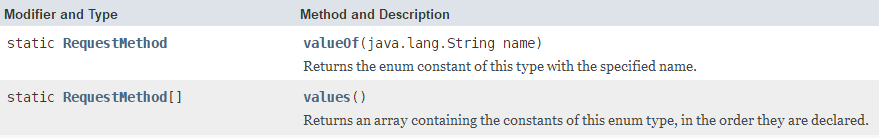
DispatcherServlet默认不支持trace和options请求。

Note that, **by default**, [DispatcherServlet](https://docs.spring.io/spring/docs/5.0.5.BUILD-SNAPSHOT/javadoc-api/org/springframework/web/servlet/DispatcherServlet.html) supports **GET, HEAD, POST, PUT, PATCH and DELETE** only. DispatcherServlet will process TRACE and OPTIONS with the default HttpServlet behavior **unless explicitly told to dispatch those request types** as well: Check out the "dispatchOptionsRequest" and "dispatchTraceRequest" properties, switching them to "true" if necessary.

## 枚举元素（8个）Enum Constants

DELETE 、GET 、HEAD 、OPTIONS 、PATCH 、POST 、PUT 、TRACE

## 2个静态方法



# Ant风格URL支持的3种风格

## Ant风格资源地址支持3种匹配符介绍

### ? : 匹配文件名中的一个字符；

### \*：匹配文件名中的任意字符，匹配1层目录；

### \*\*:匹配任意多层目录，可以为0层、1层或多层目录；

## 示例



# @PathVariable：用来获取请求路径（url）中的动态参数

## REST风格的URL

**带占位符的 URL** 是 Spring3.0 新增的功能,该功能在SpringMVC 向 **REST 目标**挺进发展过程中具有里程碑的意义。

REST风格简单描述：

传统URL：[http://www.baidu.com?**user=zhaohong&password=zhao12**](http://www.baidu.com?user=zhaohong&password=zhao12)

REST风格：[http://www.baidu.com/**zhaohong/zhao12**](http://www.baidu.com/zhaohong/zhao12)

## 映射信息：占位符

SpringMVC配置映射信息，需要在映射信息后面对应的位置加入**占位符{xxx}**。

如：@RequestMapping(value="www/baidu/com/{user}/{passwd}"}

## @PathVariable

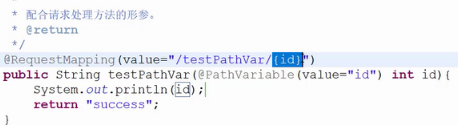
**@PathVariable**功能：@Pathvariable注解绑定它传过来的值到方法的参数上，用于将请求URL中的模板变量映射到功能处理方法的参数上，即取出uri模板中的变量作为参数。

@PathVariable绑定URI模板变量值；

@PathVariable是用来获得请求url中的动态参数的；

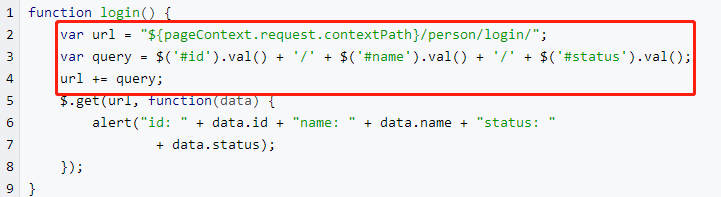
@PathVariable用于将请求URL中的模板变量映射到功能处理方法的参数上。

通过 **@PathVariable** 可以将 **URL 中占位符参数**绑定到**控制器处理方法的入参**中:URL 中的 **{xxx} 占位符**可以通过**@PathVariable(value="xxx")** 绑定到**操作方法的入参（形参）**中。如果只有value一个属性，**value**往往也可以省略。





示例2：



/\*\*

\* **@RequestMapping(value = "user/login/{id}/{name}/{status}")** 中的 {id}/{name}/{status}

\* 与 **@PathVariable int id、@PathVariable String name、@PathVariable boolean status**

\* 一一对应，按名匹配。

\*/

@RequestMapping(value = "user/login/{id}/{name}/{status}")

@ResponseBody

//@PathVariable注解下的数据类型均可用

public User login(@PathVariable int id, @PathVariable String name, @PathVariable boolean status) {

//返回一个User对象响应ajax的请求

return new User(id, name, status);

}

# @RequestParam

## 在SpringMVC后台控制层获取参数的方式主要有两种:

**一种是request.getParameter("name")，另外一种是用注解@RequestParam直接获取**。

## @RequestParam注解主要有哪些参数属性：

### value：参数名字，即入参的请求参数名字，

如username表示请求的参数区中的名字为username的参数的值将传入；

### required：是否必须，默认是true，表示请求中一定要有相应的参数，否则将报404错误码；

### defaultValue：默认值，表示如果请求中没有同名参数时的默认值，例如：

public List<EasyUITreeNode> getItemTreeNode(@RequestParam(value="id",defaultValue="0")long parentId)

# @RequestParam与@PathVariable的区别

# @Service

# @Component

# @Respository