**控制器的请求处理方法的签名可以将任何事物作为参数，我们可以将HttpServletRequest、HttpServletResponse、 String或者数字参数传递进来,这些参数可以对应请求中的查询参数、cookie值、HTTP请求头的值或其他一些可能的选项。   
  
     下面一些例子来说明**

1. 使用@RequestParam绑定请求参数值
2. @RequestMapping(value="/method1")
3. public String method1(
4. @RequestParam(value="userName",required = false) String userName,
5. @RequestParam("pwd") String pwd){
6. return "methodSignature";

}

1. 使用@CookieValue绑定请求中的Cookie值
2. @RequestMapping(value="/method2")
3. public String method2(
4. @CookieValue(value="sessionId",required=false)String sessionId){
6. return "methodSignature";

}

1. 使用@RequestHeader绑定请求报文头的属性值
2. @RequestMapping(value="/method3")
3. public String method3(
4. @RequestHeader("Accept-Encoding") String encoding,
5. @RequestHeader("Keep-Alive") long keepAlive){
6. return "methodSignature";

}

1. 使用@PathVariabl获取路径中传递参数
2. public ModelAndView helloWorld(
3. @PathVariable String id,
4. @PathVariable String str) {
5. System.out.println(id);
6. System.out.println(str);
7. return new ModelAndView( " /helloWorld " );

}

1. 使用表单对象绑定请求参数值
2. @RequestMapping(value="/method4")
3. public String method4(User user){
4. return "methodSignature";

}

1. 使用Servlet API对象作为入参   
   * 同时使用HttpServletRequest、HttpServletResponse、HttpSession
   * @RequestMapping(value="method5")
   * public void method5(
   * HttpServletRequest request,
   * HttpServletResponse response,
   * HttpSession session){
   * String userName = WebUtils.findParameterValue(request, "userName");
   * response.addCookie(new Cookie("userName",userName));

}

* + 使用Servlet原生API接口，如WebRequest和NativeWebRequest等
  + @RequestMapping(value="method6")
  + public String method6(WebRequest request){
  + String userName = request.getParameter("userName");
  + return "methodSignature";

}

* + 使用IO对象作为入参    
    如java.io.InputStream/java.io.Reader 及java.io.OutputStream/java.io.Writer
  + @RequestMapping(value="method7")
  + public void method7(OutputStream os) throws IOException{
  + Resource res = new ClassPathResource("/image.jpg");//读取类路径下的图片文件
  + FileCopyUtils.copy(res.getInputStream(), os);

}

* + 使用其他类型的参数   
    java.util.Locale/java.security.Principal   
    也可以通过Servlet的HttpServletRequest的getLocale()及getUserPrincipal()
  + @RequestMapping(value="method8")
  + public String method8(Locale locale,Principal principal){
  + return "methodSignature";

}

以上转自 ：http://blog.csdn.net/jingyuan\_/article/details/9465415   
                  http://blog.csdn.net/yaerfeng/article/details/23658391

[已有 **0** 人发表留言，猛击->> **这里**<<-参与讨论](http://lizhuquan0769.iteye.com/blog/2082478#comments)   
  
  
ITeye推荐 

* [—软件人才免语言低担保 赴美带薪读研！—](http://www.iteye.com/clicks/433)

**相关 [spring springmvc 方法] 推荐：**

[**【spring】springmvc接收请求参数的方法**](http://itindex.net/detail/50082-spring-springmvc-%E6%96%B9%E6%B3%95)

- - 开源软件 - ITeye博客

     控制器的请求处理方法的签名可以将任何事物作为参数，我们可以将HttpServletRequest、HttpServletResponse、 String或者数字参数传递进来,这些参数可以对应请求中的查询参数、cookie值、HTTP请求头的值或其他一些可能的选项. 使用@RequestParam绑定请求参数值. 使用@CookieValue绑定请求中的Cookie值. 使用@RequestHeader绑定请求报文头的属性值. 使用@PathVariabl获取路径中传递参数. 使用Servlet API对象作为入参. 同时使用HttpServletRequest、HttpServletResponse、HttpSession .

[**spring与mybatis四种整合方法**](http://itindex.net/detail/50853-spring-mybatis-%E6%96%B9%E6%B3%95)

- - 企业架构 - ITeye博客

  1、采用数据映射器（MapperFactoryBean）的方式，不用写mybatis映射文件，采用注解方式提供相应的sql语句和输入参数.   （1）Spring配置文件：.      <!-- 引入jdbc配置文件 -->.       <!--创建jdbc数据源 -->.       <!-- 创建SqlSessionFactory，同时指定数据源-->.       <!--创建数据映射器，数据映射器必须为接口-->. （2）数据映射器UserMapper,代码如下:.  (3) dao接口类UserDao，代码如下：. (4)dao实现类UserDaoImpl2,，代码如下:.

[**Spring的WebServiceTemplate访问WebService的方法及其本质原理**](http://itindex.net/detail/50920-spring-webservicetemplate-%E8%AE%BF%E9%97%AE)

- - 行业应用 - ITeye博客

WebService客户端调用的本质就是将SAOP格式的XML通过通信协议发送到WebService的服务器端,然后接收服务器端返回的XML.. 本文简单介绍一下如何通过Spring提供的WebServiceTemplate访问 Webservice,WebServiceTemplate与调用webservice的客户端已及webservice服务器端示意图如下(图片来源 于Spring in Action):. 这里以SOAP over HTTP为例,开发步骤如下:. 1,在Spring的配置文件中配置WebServiceTemplate,最简单的配置如下:.  \*这种配置省略了MessageFactory和messageSender的配置,Spring默认会使用SaajSoapMessageFactory和HttpUrlConnectionMessageSender.等同于下面的配置.

[**Spring配置文件location的几种设置方法**](http://itindex.net/detail/51494-spring-%E6%96%87%E4%BB%B6-location)

- - ITeye博客

1.默认location 默认会去加载WEB-INF下的applicationContext.xml文件，如果该文件不存在，则会抛出以下的异常. web.xml中通过servlet name自定义. 通过以下的定义，会去加载WEB-INF下面的test-servlet.xml作为spring的配置文件. 3.web.xml中通过DispatcherServlet的init-param自定义. 通过以下的定义，会去加载src/config文件夹下的test-servlet.xml作为spring的配置文件. 4.web.xml中通过ContextLoaderListener自定义. 通过以下的定义，会去加载WEB-INF文件夹下的test-servlet.xml作为spring的配置文件.

[**Spring AOP进行日志记录，管理 （使用Spring的拦截器功能获取对action中每个方法的调用情况，在方法调用前**](http://itindex.net/detail/47044-spring-aop-%E6%97%A5%E5%BF%97)

- - 行业应用 - ITeye博客

原文地址： http://hi.baidu.com/wolf\_childer/item/f0b5b0e664252cacc10d75d1.        在java开发中日志的管理有很多种. 我一般会使用过滤器，或者是Spring的拦截器进行日志的处理. 如果是用过滤器比较简单，只要对所有的.do提交进行拦截，然后获取action的提交路径就可以获取对每个方法的调用. 使用过滤器的好处是可以自己选择性的对某一些方法进行过滤，记录日志.              另外一种就是使用Spring的AOP了. 这种方式实现起来非常简单，只要配置一下配置文件就可以了. 可是这种方式会拦截下所有的对action的每个操作.

[**转: 当spring 容器初始化完成后执行某个方法**](http://itindex.net/detail/49450-spring-%E5%88%9D%E5%A7%8B-%E6%96%B9%E6%B3%95)

- - 企业架构 - ITeye博客

转自: http://www.cnblogs.com/rollenholt/p/3612440.html. 在做web项目开发中，尤其是企业级应用开发的时候，往往会在工程启动的时候做许多的前置检查. 比如检查是否使用了我们组禁止使用的Mysql的group\_concat函数，如果使用了项目就不能启动，并指出哪个文件的xml文件使用了这个函数. 而在Spring的web项目中，我们可以介入Spring的启动过程. 我们希望在Spring容器将所有的Bean都初始化完成之后，做一些操作，这个时候我们就可以实现一个接口：. //需要执行的逻辑代码，当spring容器初始化完成后就会执行该方法. 同时在Spring的配置文件中，添加注入：.

[**SpringMVC传参**](http://itindex.net/detail/48339-springmvc)

- - 企业架构 - ITeye博客

Spring MVC 的请求参数获取的几种方法. 通过@PathVariabl注解获取路径中传递参数. 用@ModelAttribute注解获取POST请求的FORM表单数据. 直接用HttpServletRequest获取. 用注解@RequestParam绑定请求参数a到变量a. 当请求参数a不存在时会有异常发生,可以通过设置属性required=false解决,. 例如: @RequestParam(value="a", required=false). 已有 0 人发表留言，猛击-> 这里<<-参与讨论. —软件人才免语言低担保 赴美带薪读研.

[**SpringMVC+ajaxfileupload上传**](http://itindex.net/detail/51136-springmvc-ajaxfileupload-%E4%B8%8A%E4%BC%A0)

- - CSDN博客互联网推荐文章

看这篇的文章之前最好看一下上篇文章这样可以更好的理解. 整个项目的基本配置和上面差不多. 不同的是在webRoot文件夹下的js中引入jQuery.js 和ajaxfileupload.js. 如何没有这个两个js文件可以到各自的官网下载. DemoController.java   跳转到upload.jsp. // 设置响应给前台的数据格式. // 设置响应给前台内容的PrintWriter对象. // 这里实现文件上传操作用的是commons.io.FileUtils类,它会自动判断/upload是否存在,不存在会自动创建. out.print("1`请选择文件后上传");.

[**模板方法**](http://itindex.net/detail/40909-%E6%A8%A1%E6%9D%BF%E6%96%B9%E6%B3%95)

- - 博客园\_首页

由于前两天刚好用到模板方法这个模式，而且这个模式相对来 比较简单实用，就写写个人的一些认知吧. 大家对宋丹丹和赵本山的小品里有一个很经典的台词一定不会陌生，而且还日常中经常引用：. 《钟点工》中宋丹丹问要把大象装冰箱，总共分几步. 赵本山就懵了，大象那么大，冰箱那么小，怎么才能把大象装冰箱里呢. 答案也很经典：三步：第1步，把冰箱门打开；第2步，把大象装进去；第3步，把冰箱门带上. 　　这个小品名字可能没多少人还记得，但是这个故事我想看过的人都会记得吧. 那么从软件设计的角度如何看上面的问题呢. 其实管他是大象，小象任何东西，只要是向冰箱里面放，最后抽象出来就三步而已，就是答案给出的三步第1步，把冰箱门打开；第2步，把大象装进去；第3步，把冰箱门带上.

[**SpringMVC 注解配置**](http://itindex.net/detail/44946-springmvc)

- - CSDN博客互联网推荐文章

在Spring项目开发中呢，最好是搞明白原理，其次装上Spring为eclipse开发的插件，这样会大大提高开发效率，而且减少了大量信息的记忆负担. SpringIDE插件，可自行到eclipse插件库中进行下载，还有其他Spring相关的plugin可以自行研究下. 当装好这个插件之后呢，可以根据向导创建一个简单的SpringMVC项目，大量的基本信息都可以自动生成，当然了是建立在明白原理的基础上，熟练了之后再去使用插件. 截个图感觉下，创建Spring项目的向导. 这里copy自动生成的xml讲解下. 根据xml中的注释也能很容易各个配置的作用. 项目的根容器上下文，用于Servlet和Filter的共享.