处理器除了实现 Controller 接口外，还可以继承自一些其它的类来完成一些特殊的功能。

1. 继承自AbstractController类

若处理器继承自AbstractController类，那么该控制器就具有了一些新的功能。因为AbstractController类还继承自一个父类WebContentGenerator。

WebContentGenerator类具有supportedMethods属性，可以设置支持的HTTP数据提交方式。默认支持GET、POST。

若处理器继承自AbstractController类，那么处理器就可以通过属性supportedMethods来限制HTTP请求提交方式了。例如，指定只支持POST的HTTP请求提交方式。

注意，这里的POST必须写为大写。

那就意味着该请求只能通过表单或AJAX请求方式进行提交，而不能通过地址栏、超链接、Html标签中的src方式进行提交。因为地址栏、超链接、Html标签中的src方式都是GET提交。否则，会给出请求方法不允许的405错误：

客户端浏览器常用的请求方式，及其提交方式有以下几种：

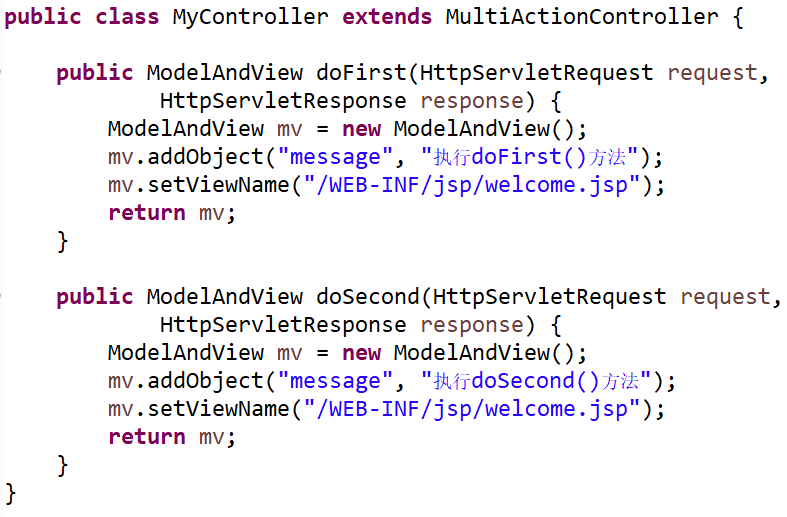


1. 继承自MultiActionController类

MultiActionController类继承自AbstractController，所以继承自MultiActionController类的子类也可以设置HTTP请求提交方式。

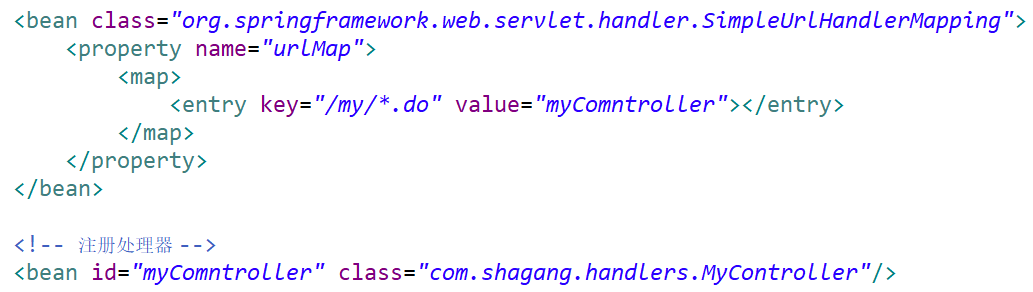
除此之外，继承自该类的处理器中可以定义多个处理方法。这些方法的签名为公共的方法，返回值为ModelAndView，包含参数HttpServletRequest与HttpServletResponse，抛出Exception异常，方法名随意。

1. **修改处理器类**



1. **修改springmvc.xml**

注意处理器类的映射路径的写法：要求必须以/xxx/\*的路径方式定义映射路径。其中\*为通配符，在访问时使用要访问的方法名代替。



1. **访问方式**

只所以通过在请求URI中写上方法名就可以访问到指定方法，是因为在MultiActionController类中有一个专门处理方法名称的解析器MethodNameResolver。该解析器作为一个属性出现，具有get与set方法。MethodNameResolver是一个接口，不同的解析器实现类，其对方法名在URI中的写法要求也是不同的。

1. **InternalPathMethodNameResolver方法名解析器（默认）**

该方法名解析器要求***方法名以URI中资源名称的身份出现***，即方法作为一种可以被请求的资源出现。也就是前面的写法：***/xxx/方法名***。

1. **PropertiesMethodNameResolver方法名解析器**

该方法名解析器中的***方法名是作为URI资源名称中的一部分出现的***，即方法名并非单独作为一种资源名称出现。例如请求时可以写为***/xxx\_doFirst***，则会访问xxx所映射的处理器的doFirst()方法。



处理器映射器的请求格式发生了改变。

另外，还需要配置一个PropertiesMethodNameResolver方法名解析器，并指定请求与要执行的方法名之间的映射关系。注意，这里的指定的请求，必须要加上.do，否则，无法完成匹配，将报404错误。

最后，将配置好的方法名解析器，注入给处理器。

1. **ParameterMethodNameResolver方法名解析器**

该方法名解析器中的***方法名作为请求参数的值出现***。例如请求时可以写为***/xxx?ooo=doFirst***，则会访问xxx所映射的处理器的doFirst()方法。其中ooo为该请求所携带的参数名，而doFirst则作为其参数值出现。

不过，打开ParameterMethodNameResolver源码，发现该类中有一个默认的参数action。即若不指定参数名称，则可以使用action作为参数。

也就是说，对于方法名称解析器ParameterMethodNameResolver的注册，可以修改为如下形式，即不指定paramName的属性值。对doFirst()方法的访问请求是：***http://localhost:8080/16-multiActionController-3/my.do?action=doFirst***