以下示例显示如何使用**Spring Web MVC**框架来实现多动作控制器的参数方法名称解析器。 MultiActionController类可在单个控制器中分别映射多个URL到对应的方法。

所下所示配置 -

package com.yiibai.springmvc;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.web.servlet.ModelAndView;

import org.springframework.web.servlet.mvc.multiaction.MultiActionController;

public class UserController extends MultiActionController{

public ModelAndView home(HttpServletRequest request,

HttpServletResponse response) throws Exception {

ModelAndView model = new ModelAndView("user");

model.addObject("message", "Home");

return model;

}

public ModelAndView add(HttpServletRequest request,

HttpServletResponse response) throws Exception {

ModelAndView model = new ModelAndView("user");

model.addObject("message", "Add");

return model;

}

public ModelAndView remove(HttpServletRequest request,

HttpServletResponse response) throws Exception {

ModelAndView model = new ModelAndView("user");

model.addObject("message", "Remove");

return model;

}

}

Java

URL映射配置文件如下 -

<bean class="com.yiibai.springmvc.UserController">

<property name="methodNameResolver">

<bean class="org.springframework.web.servlet.mvc.multiaction.ParameterMethodNameResolver">

<property name="paramName" value="action"/>

</bean>

</property>

</bean>

XML

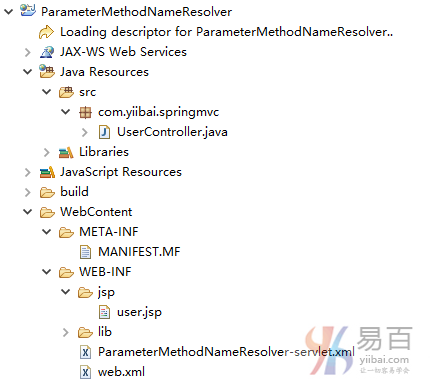
例如，使用上面的配置，如果URI：

* 对于/user/\*.html?action=home请求，DispatcherServlet将请求转发到UserController类的 home()方法。
* 对于/user/\*.html?action=add，DispatcherServlet将请求转发到UserController类的 add()方法。
* 对于/user/\*.html?action=remove，DispatcherServlet将请求转发到UserController类的 remove()方法。

首先，让我们使用Eclipse IDE，并按照以下步骤使用**Spring Web Framework**开发基于动态表单的Web应用程序：

1. 创建一个名称为 **ParameterMethodNameResolver** 的动态WEB项目。
2. 在 com.yiibai.springmvc 包下创建一个Java类UserController。
3. 在jsp子文件夹下创建一个视图文件：user.jsp。
4. 最后一步是创建所有源和配置文件的内容并运行应用程序，详细如下所述。

完整的项目文件目录结构如下所示 -



**UserController.java** 的代码如下所示 -

package com.yiibai.springmvc;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.web.servlet.ModelAndView;

import org.springframework.web.servlet.mvc.multiaction.MultiActionController;

public class UserController extends MultiActionController{

public ModelAndView home(HttpServletRequest request,

HttpServletResponse response) throws Exception {

ModelAndView model = new ModelAndView("user");

model.addObject("message", "主页面(Home)");

return model;

}

public ModelAndView add(HttpServletRequest request,

HttpServletResponse response) throws Exception {

ModelAndView model = new ModelAndView("user");

model.addObject("message", "添加(Add方法)");

return model;

}

public ModelAndView remove(HttpServletRequest request,

HttpServletResponse response) throws Exception {

ModelAndView model = new ModelAndView("user");

model.addObject("message", "删除(Remove方法)");

return model;

}

}

Java

**ParameterMethodNameResolver-servlet.xml** 的代码如下所示 -

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.0.xsd">

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="prefix" value="/WEB-INF/jsp/"/>

<property name="suffix" value=".jsp"/>

</bean>

<bean class="org.springframework.web.servlet.mvc.support.ControllerClassNameHandlerMapping">

<property name="caseSensitive" value="true" />

</bean>

<bean class="com.yiibai.springmvc.UserController">

<property name="methodNameResolver">

<bean class="org.springframework.web.servlet.mvc.multiaction.ParameterMethodNameResolver">

<property name="paramName" value="action"/>

</bean>

</property>

</bean>

</beans>

XML

**user.jsp** 的代码如下所示 -

<%@ page contentType="text/html; charset=UTF-8" %>

<html>

<head>

<title>User.jsp Page</title>

</head>

<body>

<h2>${message}</h2>

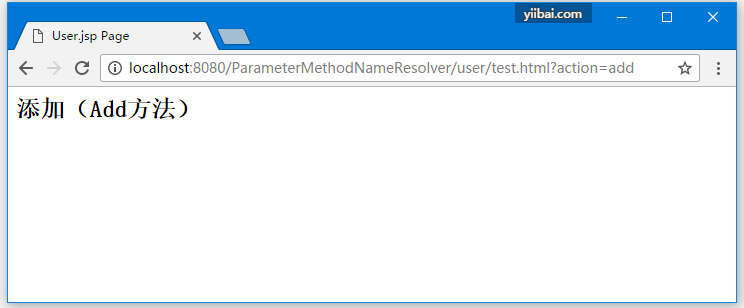
</body>

</html>

HTML

完成创建源和配置文件后，发布应用程序到Tomcat服务器。

现在启动Tomcat服务器，当访问URL => [http://localhost:8080/ParameterMethodNameResolver/user/test.html?action=add](http://localhost:8080/ParameterMethodNameResolver/user/test.html?action=add" \t "_blank) ， 如果**Spring Web**应用程序没有问题，应该看到以下结果：



当访问URL => [http://localhost:8080/ParameterMethodNameResolver/user/test.html?action=remove](http://localhost:8080/ParameterMethodNameResolver/user/test.html?action=remove" \t "_blank) ， 如果**Spring Web**应用程序没有问题，应该看到以下结果：

