其他Action

org.apache.struts.actions.DispatchAction

- DispatchAction是Action的子类，主要功能就是一个Action完成不止一种操作，例如把增删改三个操作放到一个Action中，可以减少Action类的数目
- 要在Action配置中多一个parameter属性，这个属性将指导DispatchAction找到对应的方法
- 在调用DispatchAction的时候parameter参数是不能为空的，否则会发生异常，所以必须要通过parameter参数传递方法名称
- 无需提供execute方法，但对应的方法一定要和execute方法具有相同的方法参数列表，相同的返回值类型。

创建DispatchAction示例

```
package mystruts.action;
import javax.servlet.http.*;
import org.apache.struts.action.*;
import org.apache.struts.actions.*;

public class MyDispatchAction extends DispatchAction {
    //增加数据方法
    public ActionForward addData(ActionMapping mapping, ActionForm form,
                        HttpServletRequest request, HttpServletResponse response) {
        System.out.println("增加数据");
        return null;
    }
    //修改数据方法
    public ActionForward updateData(ActionMapping mapping, ActionForm form,
                        HttpServletRequest request, HttpServletResponse response) {
        System.out.println("修改数据");
        return null;
    }
    //删除数据方法
    public ActionForward deleteData(ActionMapping mapping, ActionForm form,
                        HttpServletRequest request, HttpServletResponse response) {
        System.out.println("删除数据");
        return null;
    }
}
```

配置DispatchAction示例

- 配置Action

```
<action path="/myDispatchAction"
        type="mystruts.action.MyDispatchAction"
        scope="request"
        parameter="method"></action>
```

parameter的属性值不一定是method，这是一个自定义的值，也可以用其他的，但是一定要和调用传递的参数名一致

调用DispatchAction示例

- http://127.0.0.1:7777/mystruts/myDispatchAction.do?method=addData
- 输出：增加数据

- http://127.0.0.1:7777/mystruts/myDispatchAction.do?method=updateData
- 输出：修改数据

- http://127.0.0.1:7777/mystruts/myDispatchAction.do?method=deleteData
- 输出：删除数据

MappingDispatchAction 类

- MappingDispatchAction类是DispatchAction的子类，和DispatchAction的功能类似
- DispatchAction必需要求开发人员在请求中再带上一个参数，而MappingDispatchAction不需要这样的参数，而是需要定义多个<Action>的path和parameter属性来确定调用的方法，有几个方法就写几个action标签

创建MappingDispatchAction示例

```
package mystruts.action;
import javax.servlet.http.*;
import org.apache.struts.action.*;
import org.apache.struts.actions.*;
public class MyMappingDispatchAction extends MappingDispatchAction{
    //增加数据方法
    public ActionForward addData(ActionMapping mapping, ActionForm form,
                        HttpServletRequest request, HttpServletResponse response) {
        System.out.println("增加数据");
        return null;
    }
    //修改数据方法
    public ActionForward updateData(ActionMapping mapping, ActionForm form,
                        HttpServletRequest request, HttpServletResponse response) {
        System.out.println("修改数据");
        return null;
    }
    //删除数据方法
    public ActionForward deleteData(ActionMapping mapping, ActionForm form,
                        HttpServletRequest request, HttpServletResponse response) {
        System.out.println("删除数据");
        return null;
    }
}
```

配置MappingDispatchAction示例

- 配置Action

```
<action path="/add" type="mystruts.action.MyMappingDispatchAction"
    scope="request" parameter="addData"></action>
<action path="/update"
    type="mystruts.action.MyMappingDispatchAction"
    scope="request" parameter="updateData"></action>
<action path="/delete"
    type="mystruts.action.MyMappingDispatchAction"
    scope="request" parameter="deleteData"></action>
```

运行MappingDispatchAction示例

- http://127.0.0.1:7777/mystruts/add.do
- 输出：增加数据

- http://127.0.0.1:7777/mystruts/update.do
- 输出：修改数据

- http://127.0.0.1:7777/mystruts/delete.do
- 输出：删除数据

LookupDispatchAction类

- LookupDispatchAction类是DispatchAction类的子类，和DispatchAction类的功能类似，在一个Action中完成多个操作，主要用在一个表单中有两个提交按钮，每个按钮的功能不一样
- 按钮的value值必须是由资源文件提供
- 必须覆盖实现getKeyMethodMap方法，返回一个Map对象，这个对象存储了按钮和方法之间的映射关系
- 需要在<action>里设置parameter属性的值为按钮的name名

创建JSP示例

```
<%@ page language="java" pageEncoding="UTF-8"%>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean"%>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html"%>
<html:html lang="true">
<head>
    <title>lookup.jsp</title>
</head>
<body>
    <html:form action="/myLookupAction">
        <html:submit property="action">
            <bean:message key="lookup.button.add" />
        </html:submit>
        <html:submit property="action">
            <bean:message key="lookup.button.update" />
        </html:submit>
    </html:form>
</body>
</html:html>
```

创建ActionForm示例

```
package addressbook.form;

import org.apache.struts.action.*;

public class MyLookupForm extends ActionForm {

}
```

由于本例表单元素没有需要提交的数据，只是有两个提交按钮，所以ActionForm中没有声明任何属性

创建LookupDispatchAction示例

```
package mystruts.action;
import java.util.*;
import javax.servlet.http.*;
import org.apache.struts.action.*;
import org.apache.struts.actions.*;
public class MyLookupDispatchAction extends LookupDispatchAction {
    protected Map getKeyMethodMap() {
        Map map = new HashMap();
        map.put("lookup.button.add", "addData");
        map.put("lookup.button.update", "updateData");
        return map;
    }
    //增加数据方法
    public ActionForward addData(ActionMapping mapping, ActionForm form,
                    HttpServletRequest request, HttpServletResponse response) {
        System.out.println("增加数据");
        return null;
    }
    //修改数据方法
    public ActionForward updateData(ActionMapping mapping, ActionForm form,
                    HttpServletRequest request, HttpServletResponse response) {
        System.out.println("修改数据");
        return null;
    }
}
```

配置示例

```
<form-bean name="myLookupForm" type="mystruts.form.MyLookupForm" />

<action path="/myLookupAction"
    type="mystruts.action.MyLookupDispatchAction" scope="request"
    input="looukup.jsp" name="myLookupForm" parameter="action"/>
```

提示：parameter的属性值是自定义的，不一定是action，但是一定要和按钮的property的属性值匹配。

运行效果


lookup.jsp - Microsoft Internet Exp...
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助
后退
搜索
地址(D) http://127.0.0.1:7777/mystruts/lookup.js
转到
增加数据
修改数据
完毕
Internet