# 入门例子

## Web.xml

|  |
| --- |
| <?**xml** version=*"1.0"* encoding=*"UTF-8"*?>  <**web-app** xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/j2ee"* xmlns:web=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd http://xmlns.jcp.org/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* id=*"WebApp\_9"* version=*"2.4"*>  <**filter**>  <**filter-name**>struts2</**filter-name**>  <**filter-class**>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</**filter-class**>  </**filter**>  <**filter-mapping**>  <**filter-name**>struts2</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>  <**welcome-file-list**>  <**welcome-file**>index.jsp</**welcome-file**>  </**welcome-file-list**>  </**web-app**> |

## Action

|  |
| --- |
| **package** com.action;  **import** com.opensymphony.xwork2.ActionSupport;  **public** **class** **UserAction** **extends** **ActionSupport**{  **private** **static** **final** **long** ***serialVersionUID***=1L;  //info属性  **private** **String** info;  //info属性的getter方法  **public** **String** **getInfo**() {  **return** info;  }  //info属性的setter方法  **public** **void** **setInfo**(**String** info) {  **this**.info = info;  }  //重载execute()方法  **public** **String** **execute**() **throws** **Exception** {  info="hello,world";  **return** "hello";  }  //update()方法，更新info属性  **public** **String** **update**()**throws** **Exception**{  info="ok,updated";  **return** "update";  }  } |

和struts.xml的配置一起叫做控制器

|  |
| --- |
| <?**xml** version=*"1.0"* encoding=*"UTF-8"* ?>  <!**DOCTYPE** struts PUBLIC  "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"  "http://struts.apache.org/dtds/struts-2.0.dtd">  <**struts**>  <**constant** name=*"struts.devMode"* value=*"true"* />  <!-- 创建包default，继承struts-default -->  <**package** name=*"default"* namespace=*"/"* extends=*"struts-default"*>  <!-- 名为login的Action，实现类为LoginAction -->  <**action** name=*"user"* class=*"com.action.UserAction"*>  <!-- 返回hello.jsp -->  <**result** name=*"hello"*>/hello.jsp</**result**>  <!-- 返回 pdate.jsp-->  <**result** name=*"update"*>/update.jsp</**result**>  </**action**>  </**package**>  </**struts**> |

## 页面就懒的写了

## Struts开发模式配置

|  |
| --- |
| 下面是4个开发模式常用配置的简介---  <!-- 开启使用开发模式，详细错误提示 -->  <!-- <constant name="struts.devMode" value="true"/>-->  <!-- 指定每次请求到达，重新加载资源文件 -->  <!-- <constant name="struts.i18n.reload" value="true"/>-->  <!-- 指定每次配置文件更改后，自动重新加载 -->  <!-- <constant name="struts.configuration.xml.reload" value="true"/>-->  <!-- 指定[XSLT](https://www.baidu.com/s?wd=XSLT&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dWmH0YP1-hmyfYuhNBPW9h0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErH6zn1c1rHb3PHR4Pj0vPWR4) Result使用样式表缓存 -->  <!-- <constant name="struts.[xslt](https://www.baidu.com/s?wd=xslt&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dWmH0YP1-hmyfYuhNBPW9h0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErH6zn1c1rHb3PHR4Pj0vPWR4).nocache" value="true"/>-->  <constant name="struts.action.extension" value="do"/>  这个是配置action后缀的  如果用户需要指定多个请求后缀，则多个后缀之间以英文逗号（,）隔开。如：  <constant name="struts.action.extension" value="do,go"/> |

# 类型转换器

|  |
| --- |
| package com.action;  import java.util.List;  import com.model.User;  import com.opensymphony.xwork2.ActionSupport;  public class ListAction extends ActionSupport{  private static final long serialVersionUID = 1L;  //users属性，为List类型  private List<User> users;  //getter方法  public List<User> getUsers() {  return users;  }  //setter方法  public void setUsers(List<User> users) {  this.users = users;  }  //重载execute()方法  public String execute() throws Exception {  for (User user : users) {  System.out.println("Name:"+user.getName()+" Age:"+user.getAge()+" Tel:"+user.getTel());  }    return SUCCESS;  }  } |

这里面的元素是private List<User> users;

页面上传过来的数据形式是

|  |
| --- |
| <%@ **page** language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <%@ **taglib** prefix="s" uri="/struts-tags"%>  <!**DOCTYPE** html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <**html**>  <**head**>  <**meta** http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <**title**>Insert title here</**title**>  </**head**>  <**body**>  <**center**>  <!-- 表单 -->  <**s:form** action="list">  <!-- 表格 -->  <**table** >  <**tr**>  <**td**></**td**>  <**td**>first：</**td**>  <**td**>second:</**td**>  </**tr**>  <**tr**>  <**td**>Name:</**td**>  <!-- 设置list元素-->  <**td**><**s:textfield** name="users[0].name" theme="simple"/></**td**>  <**td**><**s:textfield** name="users[1].name" theme="simple"/></**td**>  </**tr**>  <**tr**>  <**td**>Age:</**td**>  <**td**><**s:textfield** name="users[0].age" theme="simple"/></**td**>  <**td**><**s:textfield** name="users[1].age" theme="simple"/></**td**>  </**tr**>  <**tr**>  <**td**>Tel:</**td**>  <**td**><**s:textfield** name="users[0].tel" theme="simple"/></**td**>  <**td**><**s:textfield** name="users[1].tel" theme="simple"/></**td**>  </**tr**>  <**tr**>  <**td** colspan=*"3"*>  <**s:submit**></**s:submit**>  </**td**>  </**tr**>  </**table**>  </**s:form**>  </**center**>  </**body**>  </**html**> |

在测试的时候，发现修改了struts.xml，添加了action，但是似乎没有马上生效，还是重启了之后才生效的。。

所以，遇到问题，就先重启一波把。。

# 表单提交给action

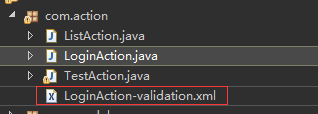
像这样可以指定提交的action名字和哪一个方法

|  |
| --- |
| <s:form action=”actionname”>  <s:submit method=”method”/>  </s:form> |

# Struts的xwork校验框架

|  |
| --- |
| **package** com.action;  **import** com.opensymphony.xwork2.ActionSupport;  **public** **class** **LoginAction** **extends** **ActionSupport**{  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  //name属性  **private** **String** name;  //age属性  **private** **int** age;  //tel属性  **private** **String** tel;  //name属性getter方法  **public** **String** **getName**() {  **return** name;  }  //name属性setter方法  **public** **void** **setName**(**String** name) {  **this**.name = name;  }  //age属性getter方法  **public** **int** **getAge**() {  **return** age;  }  //age属性setter方法  **public** **void** **setAge**(**int** age) {  **this**.age = age;  }  //tel属性getter方法  **public** **String** **getTel**() {  **return** tel;  }  //tel属性setter方法  **public** **void** **setTel**(**String** tel) {  **this**.tel = tel;  }  //重载execute方法  **public** **String** **execute**() **throws** **Exception** {  **return** "hello";  }  //log方法，Action调用的method方法  **public** **String** **log**()**throws** **Exception**{  **System**.***out***.println("log");  **return** "hello";  }    } |

在同一个包下，建立xml文件



命名规则是

Actoin名字-方法名字-validation.xml

|  |
| --- |
| <!**DOCTYPE** validators PUBLIC "-//OpenSymphony Group//XWork Validator 1.0.2//EN" "http://www.opensymphony.com/xwork/xwork-validator-1.0.2.dtd">  <**validators**>  <!-- age域 -->  <**field** name=*"age"*>  <!--域类型为int-->  <**field-validator** type=*"int"*>  <**param** name=*"min"*>10</**param**>  <**param** name=*"max"*>40</**param**>  <**message**>the age must be from 10 to 40!</**message**>  </**field-validator**>  </**field**>  <!-- name域 -->  <**field** name=*"name"*>  <!-- 域类型为requiredstring类型 -->  <**field-validator** type=*"requiredstring"*>  <**message**>the age must be from 10 to 40!</**message**>  </**field-validator**>  </**field**>  </**validators**> |

# 国际化和资源文件

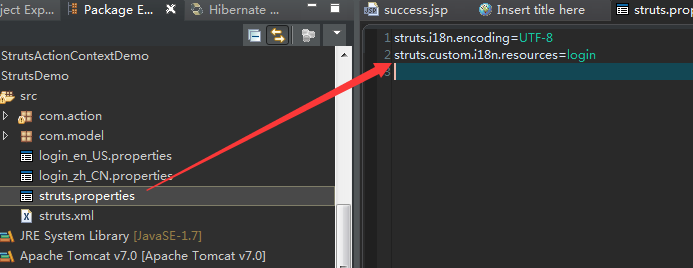
1. 包范围资源文件，只允许包下的Action访问，文件格式为package\_language\_country.properties，比如package\_zh\_CN.properties，放到包下
2. 类范围资源文件，只能被指定的Action访问，文件格式是ActionName\_language\_country.properteis
3. 全局范围资源文件，可以被所有的action和jsp访问，文件格式是baseName\_language\_country.properties,baseName自己随便定义

比如globalMessage\_zh\_CN.properties，放到src目录下

## 访问资源文件的方式

1. action通过getText方法访问
2. jsp通过<s:text>访问，指定name属性
3. 表单通过<s:textfiled>访问，指定key

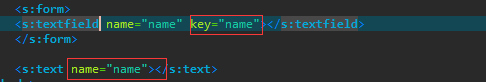
一般来说我们用全局范围资源文件就好



需要现在strtus.properties文件中定义资源文件的baseName

然后建立资源文件





在页面中引用测试一下



还不错