<%@taglib prefix="s" uri="/struts-tags"%>

×××××××××××××××××××××××××××××××××××××××××××××××××××××××

一定要记住，struts是通过set/get方法来获取属性的，一定要生成set和get

如果在标签中有id这个属性，就是将其放入pageContext中

通过ognl取得action中的值：${name}

if-elseif-else标签的使用

<s:if test="name=='zhuming'">

${ name}

</s:if>

<s:elseif test="name ==’zhum’">

${name}

</s:elseif>

迭代标签的使用

<s:iterator value="demoList" id="row" status="st">

<s:property value="#st.count"/>、<s:property value="row"/>

<s:if test="#st.odd">(奇数行)</s:if>

<s:else>(偶数行)</s:else><br/>

</s:iterator>

<h3>用法二：使用iterator标签迭代输出hashSet实例(无序):</h3>

<s:iterator value="hashSet" id="row" status="st">

<s:property value="#st.count"/>、<s:property value="row"/><br/>

</s:iterator>

<h3>用法三：使用iterator标签迭代输出treeSet实例(有序):</h3>

<s:iterator value="treeSet" id="row" status="st">

<s:property value="#st.count"/>、<s:property value="row"/><br/>

</s:iterator>

<h3>用法四：使用iterator标签迭代输出map实例:</h3>

<s:iterator value="demoMap" id="row" status="st">

<s:property value="#st.count"/>、key=<s:property value="key"/> value=<s:property value="value"/><br/>

</s:iterator>

append标签：合并多个list，set什么的

<s:append id="newlist1"> ---------------》新的list

<s:param value="demoList"/>

<s:param value="{'第二个list对象的第一个元素','第二个list对象的第二个元素'}"/>

</s:append>

<s:append id="newlist2"> --------------------》新的list list+set合并

<s:param value="demoList"/>

<s:param value="hashSet"/>

<s:param value="treeSet"/>

</s:append>

generator标签的使用方法

<s:generator separator="," val="'书法,音乐,运动'" id="mylove" count="2">

<s:iterator status=”st”>

<s:property value=”#st.count”/>,<s:property/>

</ s:iterator >

</ s:generator >

subset标签



从原籍和中介去一部分放入ValueStack栈顶，供迭代输出

<s:subset source="{'ad','asdf','asdfasdfs','easdf','sdfsfsdfs','hfadfsadf'}" count="3" start="0">

<s:iterator status="st">

<s:property value="#st.count"/>、<s:property/><br/>

</s:iterator>

</s:subset>

<h3>用法二：使用subset标签与Decider实例截取子集:</h3>

<s:bean id="mydecider" name="com.test.struts.MyDecider" />

<s:subset source="{'ad','asdf','asdfasdfs','easdf','sdfsfsdfs','hfadfsadf'}" decider="mydecider">

<s:iterator status="st">

<s:property value="#st.count"/>、<s:property/><br/>

</s:iterator>

</s:subset>

<h3>用法三：将截取到的子集存放到pageContext对象中:</h3>

<s:subset source="{'ad','asdf','asdfasdfs','easdf','sdfsfsdfs','hfadfsadf'}" count="3" start="0" id="sub"/>

<%

java.util.Iterator it = (java.util.Iterator)pageContext.getAttribute("sub");

int i = 1;

while (it.hasNext()){

out.println((i++)+"、"+it.next()+"<br/>");

}

%>

1. sort标签，将几个排序后放在栈顶



<h3>用法一：使用action标签调用tag\_action，结果不包含在本页中:</h3>

<s:action name="tag\_action" executeResult="false"/>

<h3>用法二：使用action标签调用tag\_action，并将结果包含在本页中:</h3>

<s:action name="tag\_action" executeResult="true"/>

<h3>用法三：使用action标签调用tag\_action，并阻止本页参数传入action中:</h3>

<s:action name="tag\_action" executeResult="true" ignoreContextParams="true"/>

1. action标签：直接调用指定的action

<h3>用法一：使用action标签调用tag\_action，结果不包含在本页中:</h3>

<s:action name="tag\_action" executeResult="false"/>

<h3>用法二：使用action标签调用tag\_action，并将结果包含在本页中:</h3>

<s:action name="tag\_action" executeResult="true"/>

<h3>用法三：使用action标签调用tag\_action，并阻止本页参数传入action中:</h3>

<s:action name="tag\_action" executeResult="true" ignoreContextParams="true"/>

组织了参数传入action url传值什么的

例子<http://localhost:8080/testStruts2Tag/useaction.jsp?str=src>

1. bean标签：直接创建Javabean 通过自标签对其属性进行修改



<s:bean name="com.test.struts.MyDateFormat">

<s:param name="date" value="#now"/>

格式化后的当前日期1：<s:property value="format()"/><br/>

<s:param name="pattern" value="'yyyy-MM-dd kk:mm:ss'"/>

格式化后的当前日期2：<s:property value="format()"/><br/>

<s:param name="pattern" value="'yyyy-MM-dd'"/>

格式化后的当前日期3：<s:property value="format()"/><br/>

</s:bean>

1. 国际化标签



默认的国际化消息资源文件中的test.i18n消息内容:

<s:text name="test.i18n" /> <br/>

自定义国际化消息资源文件中的test.i18n消息内容:

<s:i18n name="myMessageResource">

<s:text name="test.i18n" />

</s:i18n>

(4)Param标签：传递参数，一般与push什么的一起用，作为自标签

<s:bean name="com.test.struts.Member" id="member">

<s:param name="id" value="1"/>

<s:param name="loginName" value="'liubin'"/>

<s:param name="loginPwd" value="'123456'"/>

<s:param name="memberName" value="'刘斌'"/>

<s:param name="phone" value="'13800138000'"/>

<s:param name="address" value="'广州市天河区天河软件园'"/>

</s:bean>

<!-- 将Member对象压入ValueStack栈顶供s:property快速访问 -->

<s:push value="#member">

ID号:<s:property value="id"/><br/>

登录账号:<s:property value="loginName"/><br/>

登录密码:<s:property value="loginPwd"/><br/>

会员姓名:<s:property value="memberName"/><br/>

联系电话:<s:property value="phone"/><br/>

联系地址:<s:property value="address"/><br/>

</s:push>

(5)set标签，可以指定属性放置的作用域：scope属性

<s:bean name="java.util.Date" id="now" />

<!-- 将now对象分别存入page、request、session、application四个范围 -->

<s:set name="page\_now" scope="page" value="#now"/>

<s:set name="request\_now" scope="request" value="#now"/>

<s:set name="session\_now" scope="session" value="#now"/>

<s:set name="application\_now" scope="application" value="#now"/>

<%

out.println("page\_now="+pageContext.getAttribute("page\_now")+"<br/>");

out.println("request\_now="+request.getAttribute("request\_now")+"<br/>");

out.println("session\_now="+session.getAttribute("session\_now")+"<br/>"); out.println("application\_now="+application.getAttribute("application\_now")+"<br/>");

%>

(6)text标签，主要用于输出国际化信息

<h1>text标签测试</h1><hr/>

使用方法一：直接输出国际化消息<br/>

<s:text name="test.i18n" /><br/><br/>

使用方法二：输出带参数的国际化消息<br/>

<s:text name="test.welcome">

<s:param>刘斌</s:param>----------------》test.welcome={0}\uFF0C\u4F60\u597D占位符

</s:text>

(7)url标签

主要参数：action name value scheme anchor encode="true" includeParams="get"

使用方法一：不设置value与action属性，以当前页面URL为基本URL<br/>

<s:url/><br/><br/>

使用方法二：不设置value与action属性，以当前页面URL为基本URL，并携带请求参数<br/>

<s:url>

<s:param name=*"id"* value=*"1"*/>

</s:url><br/><br/>

使用方法三：设置value属性，并携带请求参数<br/>

<s:url value=*"left.jsp"*>

<s:param name=*"menuno"* value=*"3"*/>

</s:url><br/><br/>

使用方法四：设置value属性和scheme属性 ，并携带请求参数<br/>

<s:url value=*"left.jsp"* scheme=*" http"*>

<s:param name=*"menuno"* value=*"3"*/>

</s:url><br/><br/>

使用方法五：设置action属性和scheme属性 ，并携带请求参数<br/>

<s:url action=*"tag\_append"* scheme=*" http"*>

<s:param name=*"menuno"* value=*"3"*/>

</s:url><br/><br/>

使用方法六：同时设置action属性和value属性 ，并携带请求参数<br/>

<s:url action=*"tag\_append"* value=*"left.jsp"*>

<s:param name=*"menuno"* value=*"3"*/>

</s:url><br/><br/>

使用方法七：设置value属性和anchor属性<br/>

<s:url value=*"left.jsp"* anchor=*"a"*/><br/><br/>

使用方法八：设置encode属性和includeParams属性<br/>

<s:url value=*"left.jsp"* encode=*"true"* includeParams=*"get"*>

<s:param name=*"author"* value=*"'刘斌'"*/>

</s:url><br/><br/>

(8)property属性；输出数据

异步标签 ajax

××××××××××××××××××××××××××××××××××××××××××××××××××××

最重要的小事项

必须在使用到ajax标签的网页前面加上head标签: <s:head theme="ajax" />

1. a标签和submit标签



A标签实例

<s:a theme="ajax" href="./getdate.jsp" targets="mydiv"

loadingText="请稍候..." errorText="请求失败!" executeScripts="true">读取服务端系统 时间

</s:a>

Submit标签实例

<form id="loginform">

登录帐号：<input type="text" name="loginname" />&nbsp;&nbsp;

登录密码：<input type="password" name="loginpwd" />&nbsp;&nbsp;

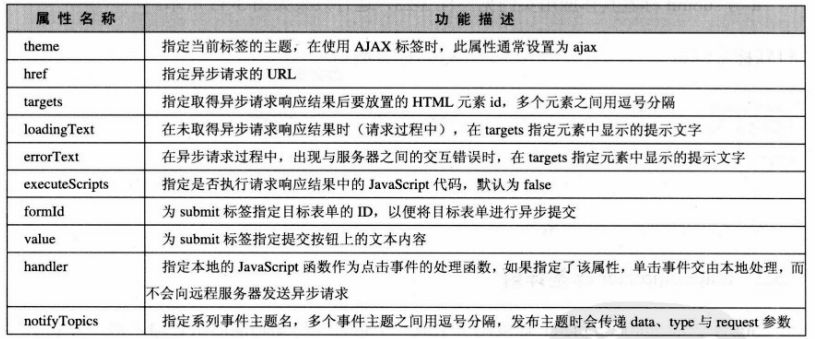
</form>

<s:submit formId="loginform" theme="ajax" href="./ajaxlogin.jsp"

targets="mydiv" loadingText="请稍候..." errorText="请求失败!"

executeScripts="true" value="异步登录" />

主要属性



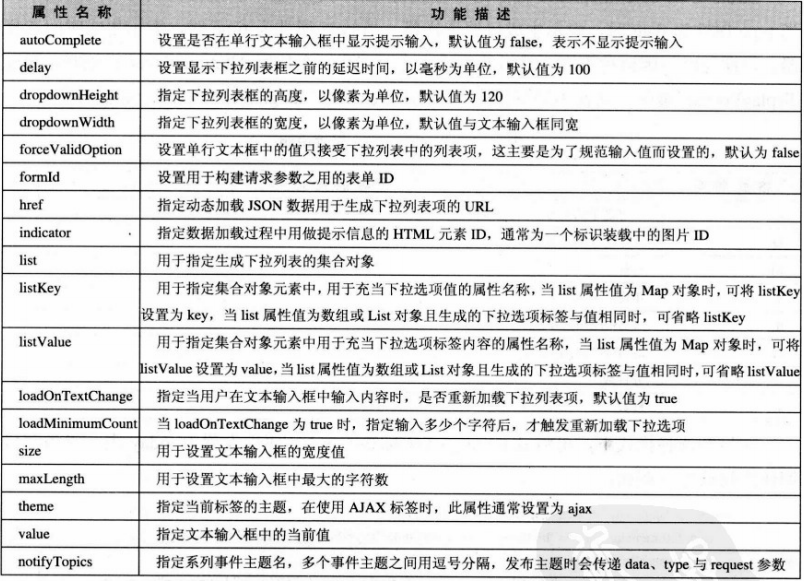
(2)Autocompleter 动态下拉列表 按下enrter键将会出现提示

<s:autocompleter name="province4" href="./getjson.jsp" theme="ajax"

indicator="myindicator" autoComplete="false" />

个人感觉上面这个是最重要的

主要属性



(3)时间标签

<s:datetimepicker name="birthday" displayFormat="yyyy-MM-dd"

value="today" /> 日期选择

<s:datetimepicker name="starttime" type="time" displayFormat="HH:mm"

language="us\_en" /> 时间选择

(4)div标签，主要用于异步加载服务器资源，并显示在div中

<s:div theme="ajax" href="./getdate.jsp" indicator="myindicator"

loadingText="请稍候..." errorText="请求失败!"

cssStyle="background-color:#dddddd">显示异步内容</s:div>

div标签用法举例2--周期性更新显示的动态内容:

<s:div theme="ajax" href="./getdate.jsp" loadingText="请稍候..."

errorText="请求失败!" updateFreq="1000"

cssStyle="background-color:#dddddd">周期更新异步内容</s:div>

div标签用法举例3--执行动态内容中的脚本:

<s:div theme="ajax" href="./getdate.jsp" executeScripts="true"

loadingText="请稍候..." errorText="请求失败!"

cssStyle="background-color:#dddddd">执行异步脚本</s:div>

1. table标签页

<s:tabbedPanel id="tabpanel2" theme="ajax" cssStyle="width:400px;">

<s:div id="tab4" label="第一个Tab页" theme="ajax" cssStyle="padding:10px;"

href="./getdate.jsp" loadingText="请稍候..." errorText="请求失败!"></s:div>

<s:div id="tab5" label="第二个Tab页" theme="ajax" cssStyle="padding:10px;"

href="./getdate.jsp" updateFreq="1000" loadingText="请稍候..."

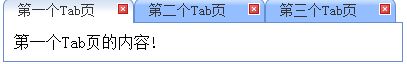
errorText="请求失败!" />

<s:div id="tab6" label="第三个Tab页" theme="ajax" cssStyle="padding:10px;"

href="./getjson.jsp" loadingText="请稍候..." errorText="请求失败!"></s:div>

</s:tabbedPanel>

异步的加载几个分页

效果图

Struts 加强对ajax的支持

这个要学习js----》jquery

其他的一些标签：用于输出action中的信息的

1. actionerror 与actionmessage

与：addActionError()和addActionMessage()连用

1. fileerror 与 filemessage

与addFileError()和addFileMessage()

1. 树形组件

 这里面的代码值得推敲