**CHƯƠNG 5. Nghiên cứu Form-tag**

1. **Giới thiệu về Tag Library trong Struts.**

* Tags Library trong Struts gồm 2 loại: UI tags và Non UI tags.
* Non UI tags giúp flow controls và data access.
* UI tags gồm 2 nhóm nhỏ:
  + Dùng trong data entry (form tags) (nghiên cứu chủ yếu form tags trong file này).
  + Dùng hiển thị lỗi do người dùng nhập (input validation)
* “form” là tag chính trong form tags, nó giúp cho việc render giống “form” bên HTML. Các form tags còn lại được xem như input element

🡪 Lợi ích của form tags:

* + - Hiển thị lỗi khi người dùng nhập sai ngay tức thì.
    - Giúp trong việc chọn lựa layout hoặc themes.

1. **Khái quát chung về tags**:

* Tags trong lập trinh web là một từ khóa, có mở và đóng (theo dạng cấu trúc XML), để mô tả nội dung và kiểu cách thể hiện nội dung đó trên web.
* Khi xài tags library phải khai báo ngay đầu JSP
  + VD: <%@ taglib prefix="s" uri="/struts-tags" %>
* Một tags có nhiều attribute, giá trị một attribute trong một tags có thể gán tĩnh, hoặc gán theo kiểu OGNL
  + Gán một giá trị tĩnh cho attribute: ví dụ label = “value”, lúc đó label có giá trị là value.
  + Gán một giá trị theo kiểu OGNL, tức đặt trong %{ và }. Ví dụ:
    - Label = “%{s}” 🡪 label có giá trị là giá trị hiện có của s
    - Hoặc : label = “%{1+5}” 🡪 label có giá trị là 6.

1. **Các attribute thông dụng**

Các attribute này hầu hết có trong bất kì tag nào thuộc UI tags

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Mô tả |
| cssClass | String | Class CSS để rendered thẻ |
| cssStyle | String | Style CSS để rendered thẻ |
| title | String | Chỉ rõ thuộc tính title của HTML |
| disable | String | Chỉ rõ thuộc tính disabled của HTML |
| label\* | String | Đặc tả label cho một form element trong xhtml và ajax theme |
| labelPostion\* | String | Đặc tả vị trí label trong xhtml và ajax theme, mặc định giá trị này là top và left |
| key | String | Tên thuộc tính trong field input đại diện. Đây là một shortcut name và cũng là thuộc tính label |
| requiredposition\* | String | Đặc tả vị trí label “required” của form element trong xhtml và ajax theme. Giá trị mặc định là left và right. |
| name | String | Đặc tả thuộc tính name HTML để một input element được map với thuộc tính trong action |
| required\* | Boolean | Trong xhtml theme thuộc tính này chỉ rõ có hay không một asterisk được add trong label |
| tabindex | String | Đặc tả thuộc tính tabindex HTML |
| value | String | Đặc tả giá trị của thẻ form |

Trong đó các attribute **name, value, label, key** là các attributre quan trọng.

Ngoài ra còn có một số attribute khác liên quan theme, javascript, tooltip như:

|  |  |  |
| --- | --- | --- |
| Table 5.2 – Template-related attributes | | |
| Name | Type | Mô tả |
| templateDir | String | Thự mục mà chứa template |
| theme | String | Tên của theme |
| template | String | Tên của template |

|  |  |  |
| --- | --- | --- |
| Table 5.3 Javascript-related attributes | | |
| Name | Type | Mô tả |
| onclick | String | Javascript onclick attribute |
| ondbclick | String | Javascript ondbclick attribute |
| onmousedown | String | Javascript onmousedown attribute |
| onmouseup | String | Javascript onmouseup attribute |
| onmouseover | String | Javascript onmouseover attribute |
| onmouseout | String | Javascript onmouseout attribute |
| onfocus | String | Javascript onfocus attribute |
| onblur | String | Javascript onblur attribute |
| onkeypress | String | Javascript onkeypress attribute |
| onkeyup | String | Javascript onkeyup attribute |
| onkeydown | String | Javascript onmousedown attribute |
| onselect | String | Javascript onselect attribute |
| onchange | String | Javascript onchange attribute |

|  |  |  |
| --- | --- | --- |
| Table 5.4 – Tooltip-related attributes | | |
| Name | Type | Mô tả |
| tooltip | String | Text được dùng trong tooltip |
| tooltiplconPath | String | Đường dẫn icon của tooltip. Giá trị mặc định là /struts/static/tooltip/tooltip.gif |
| tooltipDelay | String | Độ delay (tính mili giây) của thời gian chuột di chuyển vào vùng tooltip icon đến khi tooltip được hiện. Giá trị mặc định là 500 |

* List, listkey, listvalue là những thuộc tính quan trọng của các tag radio, checkboxlist, combobox, select, double select
* Tùy thuộc cấu trúc mỗi tag mà các thuộc tính này sẽ nhận các kiểu đầu vào khác nhau (ví dụ radio tag không có attribute value…)
* Nếu sử dụng các thuộc tính này thì attribute list là bắt buộc phải có, listkey và listvalue có thể kèm theo.
* Ta có thể dùng đối tượng kiểu String, mảng, java.until.Enumeration, java.until.Iterator, java.until.Map, hoặc Collection gán vào attribute list. Các đối tượng này có thể nằm trong action object, session object, hoặc ServletContext object.
  + Đối tượng dạng String (mảng String):
    - *Ví dụ:* <s:select list="{'Atlanta', 'Chicago', 'Detroit'}"/>.
    - Lúc này tag select sẽ có dạng cấu trúc code khác tương đương như:

<select>

<option value="Atlanta">Atlanta</option>

<option value="Chicago">Chicago</option>

<option value="Detroit">Detroit</option>

</select>

* + - 🡪 Attribute value và label của tag select sẽ giống nhau, nếu chúng ta muốn value và label của tag select khác nhau thì làm như sau:

<s:select list="#{'1':'Atlanta', '2':'Chicago', '3':'Detroit'}"/>

* + - Đoạn mã trên sẽ tương đương với:

<select>

<option value="l">Atlanta</option>

<option value="2">Chicago</option>

<option value="3">Detroit</option>

</select>

* + Đối tượng Map: được dùng trong các trường hợp attribute value khác label. Sử dụng Map rất đơn giản, ta sẽ nhập giá trị Map key và label của Map value. Ví dụ:

Map<Integer, String> cities = new HashMap<Integer, String>();

cities.put(1, "Atlanta");

cities.put(2, "Chicago");

cities.put(3, "Detroit");

* + - Nếu “cities” là một action properties, thì gán vào list ở jsp sẽ:

<s:select list="cities"/>

* + - Hoặc “cities” là một appliaction attribute, thì đoạn mã gán ở jsp sẽ:

<s:select list="#application.cities"/>

* + Đối tượng Collection hoặc mảng đối tượng: list, listKey và listValue sẽ được sử dụng trong tag, ví dụ ta có 1 List là “cities”, từng đối tượng trong List này có id và name properties. Khi đó đoạn mã trong jsp là:

<s:select list="cities" listKey="id" listValue="name" />

1. **Tag form**

* Như đã nói tag form là tag khá quan trọng, nó dùng để render như <form> bên HTML, nó gồm các attribute sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Table 5.5 Form tag attributes | | | |
| Name | Data Type | Default Value | Mô tả |
| acceptcharset | String |  | Comma hoặc giới hạn không gian bộ kí tự cho phép trong form |
| action | String | current action | Action submit của form |
| enctype | String |  | Thuộc tính enctype của form |
| method | String | post | Phương thức của form |
| onsubmit | String |  | Thuộc tính onsubmit của javascript |
| openTemplate | String |  | Template được dùng để rendere form |
| portletMode | String |  | Chế độ porlet để hiển thị sau khi form submit |
| target | String |  | Thuộc tính targer form |
| validate | Boolean | false | Chỉ rõ validation ở phía client-side được thực hiện trong xhtml/ajax theme |
| windowState | String |  | Trạng thái window thể hiện sau khi formsubmit |

- Ví dụ về form tag:

<s:form>

...

</s:form>

- Với đoạn mã trên sẽ tương đương đoạn mã sau dưới dạng HTML  
<form id="..." name="..." method="POST"

action="..." onsubmit="return true;">

<table class="wwFormTable">

</table>

</form>

1. **Các tag thông dụng khác:**

* Textfield: là một loại tag render input text filed.
* Password: được kế thừa từ textfield và có thêm thuộc tính showPassword. Ví dụ

<s:form action="Product\_save">

<s:password key="password" showPassword="true"/>

. . .

</s:form>

* Hidden: render hidden field.
* Submit tag: là dạng submit button, gồm 3 dạng:
  + Input : <input type =“submit”…./>
  + Button: <button type =“submit”…./>
  + Image: <input type =“image”…./>
* Reset tag: có dạng reset button, gồm có 2 loại, dựa vào attribute type:
  + Input: <input type =”reset”…/>
  + Button: <button type =”reset”…/>
* Head tag: tương tự như thể head bên HTML, đa số thẻ head hay được áp dụng cho AJAX.
* Textarea tag: là dạng textarea, gồm các attribute chính là:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Default Value | Description |
| cols | integer |  | Thuôc tính cols HTML |
| readonly | boolean | false | Chỉ rõ textarea có chỉ đọc hay không |
| rows | integer |  | Thuộc tính rows HTML |
| wrap | boolean |  | Thuộc tính wrap HTML |

* Checkbox tag: là dạng checkbox đơn, mặc định của attribute value là true hoặc false, và tag có một attribute là:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Default Value | Description |
| fieldValue | String | true | Giá trị thật sự checkbox |

* + FieldValue là attribute dùng để thay đổi giá trị value khác mặc định để gửi về server khi submit

<s:checkbox key=”checkbox1” value=”Y” fieldValue=”Y”/>

* + Tuy nhiên có một vấn đề là server không biết một checkbox trên form có được check hay không. Bởi vì khi value là “on” (đã check) thì nó sẽ gọi và gán value là true, nhưng khi value là “off” (không check), thì nó sẽ không gọi, nếu trước đó value là true thì không check vẫn là true.
  + Do đó để khắc phục lỗi này, checktag sẽ tự có một hidden tag đi kèm (tức có 2 attribute value mỗi tag sẽ lưu trạng thái của checkbox). Nếu check thì sẽ gửi lên server 2 giá trị true của 2 tag này, nhưng nếu không check, thì chỉ một mình giá trị hidden được gửi lên server.
  + Checkbox tag cũng có thể dùng để hiển thị một list danh sách (tự động sinh nhiều checkbox tag tương ứng danh sách), hoặc cũng có thể dùng checkbox list tag
* Radion tag: render ra một nhóm radio button. Số lượng các radio button này phụ thuộc số lượng mà ta đã gán vào list attribute của tag.
* Select tag: render ra kiểu chọn lựa các phần tử (giống combobox của .NET). Tag này có một số attribute quan trọng sau:
  + Chú ý: một **phần tử** trong Select tag được gọi là **option**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Default Value | Description |
| emptyOption | Boolean | false | Add thêm một option trống lên đầu header |
| headerKey | String |  | Key cho phần tử đầu tiên trong list |
| headerValue | String |  | Value cho phần tử đầu tiên trong list |
| list\* | String |  | Mảng gốc được đưa vào select tag |
| listkey | String |  | Thuộc tính của đối tượng trong list mà hỗ trợ cho các giá trị của option |
| listValue | String |  | Thuôc tính của đối tượng trong list mà hỗ trợ cho các label của option |
| multiple | boolean | false | Cho phép đa lựa chọn |
| size | integer |  | Số lượng option được hiển thị |

* + HeaderKey và headerValue là 2 attribute để thêm một option vào đầu. ví dụ:

<s:select name="city" label="City"

headerKey="0" headerValue="[Select a city]"

list="#{'1':'Atlanta', '2':'Chicago', '3':'Detroit'}"

/>

* Checkboxlist tag: render một nhóm các checkbox, được mapped từ mảng string thông qua list, listKey và listValue attribute. Nếu không có checkbox nào được check thì khi submit, server sẽ nhận được mãng rỗng (không phải null).
* Combobox tag: render bao gồm input field và select element (giống select tag). Các attribute cơ bản của tag này là:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Default Value | Description |
| emptyOption | Boolean | false | Add thêm một option trống lên đầu header |
| headerKey | String |  | Key cho phần tử đầu tiên trong list, nên là -1 |
| headerValue | String |  | Value cho phần tử đầu tiên trong list |
| list\* | String |  | Mảng gốc được đưa vào select tag |
| listkey | String |  | Thuộc tính của đối tượng trong list mà hỗ trợ cho các giá trị của option |
| listValue | String |  | Thuôc tính của đối tượng trong list mà hỗ trợ cho các label của option |
| maxlength | integer |  | Thuộc tính maxlength trong HTML |
| readonly | boolean | false | Cho phép phần tử được render là chỉ đọc hay không |
| size | integer |  | Số lượng option được hiển thị |

* + Không như select tag, các giá trị gán vào combobox không cần key. Do đó, label trong cac giá trị lựa chọn, đều không có value.
* Doubleselect tag: render ra 2 select element, mà giữa 2 select elenment này có quan hệ liên kết nhóm với nhau. Các attribute của tag này là:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Default Value | Description |
| doubleCssClass | String |  | Class Css cho phần tử select thứ 2 |
| doubleCssStyle | String |  | Style Css cho phần tử select thứ 2 |
| doubleDisabled | boolean | false | Cho phép disable phần tử thứ 2 |
| doubleEmptyOption |  | false | Cho phép thêm option trống vào element thứ 2 hay không |
| doubleHeaderKey | String |  | Header key cho phần tử select thứ 2 |
| doubleHeaderKey | String |  | Header value cho phần tử select thứ 2 |
| doubleList\* | String |  | Đối tượng mảng cho phần tử select thứ 2 |
| doubleListKey | String |  | Thuộc tính của đối tượng trong list thứ 2 mà hỗ trợ các label của option |
| doubleListValue | String |  | Thuộc tính của đối tượng trong list thứ 2 mà hỗ trợ các value của option |
| doubleMultiple | Boolean | false | Cho phép phần tử select thứ 2 được đa lựa chọn |
| doubleName\* | String |  | Name của phần tử select thứ 2 |
| doubleSize | integer |  | Số lượng option của phần tử select thứ 2 |
| doubleValue | String |  | Giá trị cho phần tử select thứ 2 |
| emptyOption | Boolean | false | Cho phép thêm một option trống vào phần tử select đầu. |
| formName | String |  | Name của form đang chứa. |
| headerKey | String |  | Header key cho phần tử select đầu tiên |
| headerValue | String |  | Header value cho phần tử select đầu tiên |
| list |  |  | Đối tượng mảng cho phần tử select đầu tiên |
| listKey | String |  | Thuộc tính đối tượng trong list đầu mà hỗ trợ các giá trị option |
| listValue | String |  | Thuộc tính đối tượng trong list đầu mà hỗ trợ các label option |
| multiple | String |  | Cho phép phần tử select thứ 1 được đa lựa chọn |
| size | Integer |  | Số lượng option của phần tử select thứ 1 |

* + Ví dụ: class DoubleSelectTestAction là một action class có 2 properties là country và city liên kết tới doubleselect tag trong DoubleSelect.jsp

Listing 5.24 Class DoubleSelectTestAction

|  |
| --- |
| package app05a;  import com.opensymphony.xwork2.ActionSupport;  public class DoubleSelectTestAction extends ActionSupport {  private String country;  private String city;  // getters and setters not shown  } |

Listing 5.25 Trang DoubleSelect.jsp

|  |
| --- |
| <%@ taglib prefix="s" uri="/struts-tags" %>  <html>  <head>  <title>doubleselect Tag Example</title>  <style type="text/css">@import url(css/main.css);</style> <style>  select {  width:170px;  }</style>  </head>  <body>  <div id="global" style="width:300px">  <s:form>  <s:doubleselect label="Select Location"  name="country"  list="{'US', 'Canada', 'Mexico'}"  doubleName="city"  doubleList="top == 'US' ?  {'Atlanta', 'Chicago', 'Detroit'}  : (top == 'Canada' ?  {'Vancouver', 'Toronto', 'Montreal'}  : {'Mexico City', 'Tijuana'})"  />  <s:submit/>  </s:form>  </div>  </body>  </html> |

* + Lúc đó ta sẽ có kết quả là:



* Themes: Mỗi tag trong UI tag sẽ được render thành một hoặc nhiều element HTML. Trong struts ta có thể chọn cách render từ các tag ra các element HTML khác nhau. Ví dụ, mặc định của form tag trong JSP sẽ render ra gồm form element và table element trong HTML

<s:form></s:form>

Sẽ render thành:

<form id="..." name="..." onsubmit="return true;" action="..."

method="post">

<table class="wwFormTable">

</table>

</form>

* + Tuy nhiên, đôi khi ta muốn các tag trong JSP render thành những element HTML khác (gọi là template), ví dụ form tag render ra chỉ có form element mà không có table element HTML. Thì ta có thể chọn loại template khác (nằm trong FreeMarker).
  + Những template giống nhau được đặt trong cùng một **theme.** Có 4 loại theme:
    - Simple: Các template trong simple theme này sẽ render các UI tags thành những element HTML đơn giản nhất, không có label attribute. Ví dụ <s:form> sẽ được render thành form element, không có table element.
    - Xhtml: đây là theme mặc định, các template trong theme này khi render thường có table element. Ví dụ <s:form> render thành form element và table element.
    - Css\_xhtml: theme này tương tự xhtml nhưng khi layout có sử dụng css
    - Ajax: theme này dựa trên xhtml, nhưng có cung cấp các đặc tính nâng cao của ajax.
  + Cách khai báo sử dụng theme cho từng tag, phụ thuộc vào theme attribute mỗi tag.

<s:textfield theme="simple" name="userId"/>

* + Nếu theme attribute không được sử dụng trong tag, thì tag đó mặc định sửa dụng theme của form tag.

<s:form theme="css\_xhtml">

<s:checkbox theme="simple" name="daily" label="Daily news alert"/>

<s:checkbox name="weekly" label="Weekly reports"/>

<s:checkbox theme="simple" name="monthly" label="Monthly reviews"

value="true" disabled="true"

/>

<s:submit/>

</s:form>