Struts Code Peaces – <html:select> Element

We explain the Struts <html:select> element and illustrate the usage with some small examples.

Generals

Author:
Sascha Wolski
http://www.laliluna.de/tutorials.html – Tutorials für Struts, EJB, xdoclet und eclipse.

Date:
February 22th 2005

The <html:select> element

The *<html:select>* renders a HTML *<select>* element. A property of a form bean is associated with this element, which holds the value if the element is submitted to the server

The *<html:select>* element works in two different modes:

- Single selection of an element
- Multiple selection of elements

The following example shows the source code of the jsp file.

```
<html:select property="selectedItem">
      <html:option value="Maria">Maria</html:option>
      <html:option value="Klaus">Klaus</html:option>
</html:select>
```

The following HTML source code is rendered at runtime.

```
<select name="selectedItem">
      <option value="1">Marie</option>
      <option value="2">Klaus</option>
</select>
```

Attributes of the <html:select> element

Now the important attribute will be explained. You find a complete list of all available attributes for this tag in the API of the HTML tag library.

Name	Beschreibung
disabled	Disable the element (true / false)
multiple	If true, the user can select multiple options.
name	Name of the bean which contain the properties.
property	Name of the property or request parameter, which is associated with this element.
size	Size of the displayed elements of the select element
value	The value to compare with for marking an option selected.

Usage of the <html:select> element

Create a new struts project to illustrate the usage of the *<html:select>* element.

Create a new form bean

Create a a action form class *ExampleForm* in the package *de.laliluna.tutorial.select.form*.

Define a property *selectItem* of type String, which is associated with an *<html:select>* element. This property holds the value of the associated element.

Add a second property, an array, *selectedItems* of type String. This property associated with a *<html:select>* element, which allows multiple selection of elements and holds the values of the selected elements.

The following source code shows the action form class:

```
public class ExampleForm extends ActionForm {

      //property for single selection
      private String selectedItem;
      //property for multiple selection
      private String[] selectedItems;

      public String getSelectedItem() {
            return selectedItem;
      }
      public void setSelectedItem(String selectedItem) {
            this.selectedItem = selectedItem;
      }
      public String[] getSelectedItems() {
            return selectedItems;
      }
      public void setSelectedItems(String[] selectedItems) {
            this.selectedItems = selectedItems;
      }

      /**
       * Reset method
       * @param mapping
       * @param request
       */
      public void reset(ActionMapping mapping,
                                HttpServletRequest request) {
            //resetting properties
            this.selectedItem = new String();
            this.selectedItems = new String[0];
      }
}
```

Create a new action class

Provide a action class *ExampleAction* in the package *de.laliluna.tutorial.select.action.*

Use the System.out.println() to check the values which are submit to the server.

The following source code shows the action class:

```
public class ExampleAction extends Action {

      public ActionForward execute(
            ActionMapping mapping,
            ActionForm form,
            HttpServletRequest request,
            HttpServletResponse response) {
            ExampleForm exampleForm = (ExampleForm) form;

            //debug output of selectedItem property
            System.out.println("Selected item = " +
                                exampleForm.getSelectedItem());

            //debug output of the selectedItems property
            for(int i=0;i<exampleForm.getSelectedItems().length;i++){
                  System.out.println("Multiple item (" + i +") = " +
                                      exampleForm.getSelectedItems()[i]);
            }
            return mapping.findForward("success");
      }
}
```

Create the struts-config.xml

Now open the struts-config.xml and specify the form bean and the action mapping.

```
<struts-config>
   <form-beans>
      <form-bean name="exampleForm"
type="de.laliluna.tutorial.select.form.ExampleForm" />
   </form-beans>

   <action-mappings>
      <action
         name="exampleForm"
         path="/example"
         scope="request"
         type="de.laliluna.tutorial.select.action.ExampleAction">
         <forward name="success" path="/form/example.jsp" />
      </action>
   </action-mappings>
</struts-config>
```

Create the jsp file

Create a file named *example.jsp* in the folder */WebRoot/form/.*

Open the jsp file *example.jsp* and add the following source code.

```
<%@ page language="java"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-logic" prefix="logic"%>

<html>
      <head>
            <title>example.jsp</title>
      </head>
      <body>
            <html:form action="/example">

            .... sample code ...

            </html:form>
      </body>
</html>
```

Within the *<html:form>* element add the first example.

Example 1

The first example shows a normal select element with four options. The user can select one element and submit the form. The value of the selected option is set in the associated property *selectedItem* of the form bean and can be output in the action class. The attribute *property* specifed the associated *property* of the form bean.

```
<h4>Single selection with the &lt;html:select&gt; Tag</h4>

<html:select property="selectedItem">
      <html:option value="Maria">Maria</html:option>
      <html:option value="Peter">Peter</html:option>
      <html:option value="Klaus">Klaus</html:option>
</html:select>

<html:submit/>
```

Example 2

Example 2 shows the usage of the attribute *multiple* to allow the user select multiple options. An Array of type String inside the form bean holds the values of the selected options. The attribute *property* specify the name of this array.

```
<h4>Multiple selection with the &lt;html:select&gt; Tag</h4>

<html:select property="selectedItems" multiple="true" size="3">
      <html:option value="Maria">Maria</html:option>
      <html:option value="Peter">Peter</html:option>
      <html:option value="Klaus">Klaus</html:option>
</html:select>

<html:submit/>
```

Now you can test the project. We use a jboss or tomcat installation. Call the project with the following link.

http://localhost:8080/CheckboxTag/example.do