

## Struts Code Peaces – <html:multibox> Element

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

### Generals

#### Author:

Sascha Wolski

Sebastian Hennebrueder

<http://www.laliluna.de/tutorials.html> – Tutorials für Struts, EJB, xdoclet und eclipse.

#### Date:

February 22th 2005

### The <html:multibox> element

The <html:multibox> element is used to manage multiple checkbox elements, for example in lists. The value, which will be transferred to the server from a selected checkbox, can be any value of type string. The associated property in the form-bean is an array of type string, which holds a list of selected values after submitting the form.

The following examples shows the source code of the JSP file:

```
<html:multibox property="selectedItems" value="1"> Maria  
<html:multibox property="selectedItems" value="2"> Klaus
```

The following HTML source code is rendered at runtime:

```
<input type="checkbox" name="selectedItems" value="1"> Maria  
<input type="checkbox" name="selectedItems" value="2"> Klaus
```

### Attributes of the <html:multibox> element

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

<http://struts.apache.org/userGuide/struts-html.html>

Name	Beschreibung
disabled	Disables the element (true / false)
name	Name of the bean which contains the properties.
property	Name of the property or request parameter, which is associated with this element.
value	The value which is submitted to the server. If not specified the value inside the body of this tag will be used.

### Usage of the <html:multibox> element

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

#### Create a new action class

Create a new action class *ExampleAction* in the package *de.laliluna.tutorial.multibox.action*.

We use the class *LabelValueBean* to add a label and a value to the collection.

```
public class ExampleAction extends Action {  
  
    public ActionForward execute(  
        ActionMapping mapping,  
        ActionForm form,  
        HttpServletRequest request,  
        HttpServletResponse response) {
```

```

        ExampleForm exampleForm = (ExampleForm) form;

        //initial the customers collection
        Collection customers = new ArrayList();
        customers.add(new LabelValueBean("Marie", "1"));
        customers.add(new LabelValueBean("Klaus", "2"));
        customers.add(new LabelValueBean("Peter", "3"));
        //set customer collection in the request
        request.setAttribute("customers", customers);

        return mapping.findForward("success");
    }
}

```

## Create a new form bean

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

Define a property array *selectedItem* of type String. This property holds the values of the associated multibox elements.

Provide a getter and setter method for the property.

The following source code shows the action form class:

```

public class ExampleForm extends ActionForm {

    private String[] selectedItems = {};

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

```

## Create the struts-config.xml

Open the struts-config.xml and define the form bean and action mapping.

```

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

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

```

## Create the jsp file

Create a JSP 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

There are three `<html:multibox>` elements in the first example. The attribute *property* of the element refers to the property *selectedItems* of the form bean. The attribute value specifies the value of each element, which are submitted to the server.

```

<h4>Simple use of <html:multibox> tags</h4>

<html:multibox property="selectedItems" value="A" /> Maria
<html:multibox property="selectedItems" value="K" /> Klaus
<html:multibox property="selectedItems" value="P" /> Peter
<br />
<html:submit property="btnApply"/>

```

## Example 2

The example 2 shows the usage of an `<html:multibox>` element inside an iteration `<logic:iterate>`. Inside the iteration the value of the current element of the collection is assigned to the `<html:multibox>` element. `<bean:write ..>` inside the body of the checkbox element sets the value, which is submitted to the server. The label is set with a `<bean:write ...>` after the checkbox element. The attribute *property* of the multibox element specifies the associated property *selectedItem* of the form bean.

```

<h4>Use of <html:multibox> tag inside a <logic:iterate></h4>

<logic:iterate name="customers" id="customer">
  <html:multibox property="selectedItems">
    <bean:write name="customer" property="value" />
  </html:multibox>

  <bean:write name="customer" property="label" />
</logic:iterate>

<br />
<html:submit property="btnApply"/>

```

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

<http://localhost:8080/MultiboxTag/example.do>