

Easy Java Websites (EJW)[™]

Form Tags

Copyright © 2006 - Present, EasierJava All Rights Reserved.

This Easier Java Websites Form Tags manual is a copyrighted work and is not transferable. If you received this manual somewhere other than EasierJava.com, please report it.

EasierJava and EJW are trademarks or registered trademarks of EasierJava, Inc. in the U.S. and other countries. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All other marks are trademarks or registered trademarks of their respective holders in the U.S. and other countries.

Tags Overview	3
Custom Functions	3
Contains	3
Messages	4
Input	4
Checkbox	4
Radio	5
Textarea	5
Select	6

Tags Overview

EJW provides several useful forms tags that provide "round trip" control of your form data. By "round trip" we mean that the data submitted by the user will continue to exist if the form is rejected because it doesn't pass validation.

To use EJW forms tags use:

```
<%@ taglib uri="http://com.easierjava/ejw/forms" prefix="ejw" %>
```

in your JSP. You can then access the form tags with:

```
<ejw:messages ... />  
<ejw:input ... />  
<ejw:checkbox ... />  
<ejw:radio ... />  
<ejw:textarea ... />  
<ejw:select ... />
```

You can use all EJW tags, or you can intermix EJW tags with other form tags, it doesn't matter.

EJW tags take EJW specific attributes as defined below, but they can also take attributes as would normally be used with a given form tag. For example, if you're using the HTML form tag `textarea` and you want to use different colors, you would use:

```
<textarea ... style="color:blue; background-color:#ccc;"></textarea>
```

With EJW, you can use:

```
<ejw:textarea ... attributes='style="color:blue; background-color:#ccc;"' />
```

thus passing in HTML attributes for the given tag. The benefit of the "attributes", and the input type (explained below), attributes is that EJW provides for current and future changes to HTML forms (within reason), and supports HTML 5 out of the box, while still providing "round-trip" to the server functionality.

Custom Functions

Contains

EJW provides a single custom function that provides the ability to work with collections and contains, so you can use collections (i.e. sets) in your JSTL.

Simply use:

```
<%@ taglib uri="http://com.easierjava/ejw/functions" prefix="cfn" %>
```

in your JSP. You can then use:

```
`${cfn:contains(collection, value)}`
```

which returns a true or false (boolean).

Messages

All application messages, errors, and exceptions are displayed with the following tag:

```
<ejw:messages attributes="" />
```

attributes	optional	All messages are enclosed in a HTML division tag. Attributes is associated with that division. See attributes above.
------------	----------	--

Input

Equivalent to the HTML input form tag.

Syntax:

```
<ejw:input  
  id=""  
  type=""  
  value=""  
  defaultValue=""  
  attributes="" />
```

id	required	A unique id for your tag. It should represent the property id of the value. It will be used with the HTML id and name attributes.
type	required	Type is one of the HTML form input types (eg. text, password, date, ...)
value	optional	Value overrides param and defaultValue
defaultValue	optional	defaultValue can be used to provide an initial value
attributes	optional	Attributes are the HTML tag's typical attributes such as class, style, size, etc.

Checkbox

Display a HTML checkbox.

Syntax:

```
<ejw:checkbox  
  id=""  
  defaultValue=""  
  selected=""  
  attributes="" />
```

id	required	A unique id for your tag. It should represent the property id of the value. It will be used with the HTML id and name attributes.
defaultValue	optional	defaultValue="y". defaultValue can be used to provide an initial value that is passed to the server when checked.
selected	optional	true if the checkbox is selected by default
attributes	optional	see attributes above

Radio

Display a HTML radio button.

Syntax:

```
<ejw:radio  
  id=""  
  name=""  
  value=""  
  selected=""  
  attributes="" />
```

id	required	A unique id for your tag. It should represent the property id of the value. It will be used with the HTML id attribute.
name	required	The name for the radio group
value	required	the current selected value or empty
selected	optional	true if this radio button is selected
attributes	optional	see attributes above

Textarea

Display a HTML text area.

Syntax:

```
<ejw:textarea
  id=""
  rows=""
  cols=""
  value=""
  defaultValue=""
  attributes="" />
```

id	required	A unique id for your tag. It should represent the property id of the value. It will be used with the HTML id and name attributes.
rows	optional	The number of rows to display.
cols	optional	The number of columns to display
value	optional	The value
defaultValue	optional	The defaultValue
attributes	optional	see attributes above

Select

Display a HTML select box (drop down or area).

Syntax:

```
<ejw:select
  id=""
  multiple=""
  size=""
  selectedValue=""
  selectedValues=""
  commaSeperatedValues=""
  valuesCollection=""
  valuesMap=""
  selectAttributes=""
  optionAttributes="" />
```

id	required	A unique id for your tag. It should represent the property id of the value. It will be used with the HTML id and name attributes.
multiple	optional	true or false - Allow multiple selections
size	optional	Use select area of "size", instead of drop down box.
selectedValue	optional	The selected value.

selectedValues	optional	java.util.Collection - The selected values when multiple values are selected and selectable.
commaSeperatedValues	optional*	Comma separated list of values for the select.
valuesCollection	optional*	java.util.Collection - A Java collection of values.
valuesMap	optional*	java.util.Map - A Java map with (eg. HashMap) key/value pairs. The key part is the option value and the value part is the displayed value.
selectAttributes	optional	Attributes associated with the HTML select tag. See attributes above.
optionAttributes	optional	Attributes associated with the HTML options tag. See attributes above.

*One of these must be defined