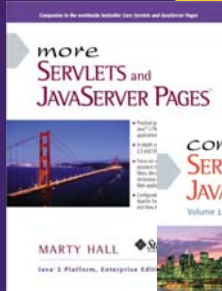




# Using JSP Custom Tag Libraries for Ajax

Originals of Slides and Source Code for Examples:  
<http://courses.coreservlets.com/Course-Materials/ajax.html>

**Customized J2EE Training:** <http://courses.coreservlets.com/>  
Servlets, JSP, Struts, JSF/MyFaces, Hibernate, Ajax, Java 5, Java 6, etc. Ruby/Rails coming soon.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



For live Ajax & GWT training, see training courses at <http://courses.coreservlets.com/>.



Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this tutorial. Available at public venues, or customized versions can be held on-site at your organization.

- Courses developed and taught by Marty Hall
  - Java 5, Java 6, intermediate/beginning servlets/JSP, advanced servlets/JSP, Struts, JSF, Ajax, GWT, custom mix of topics
- Courses developed and taught by [coreservlets.com](http://coreservlets.com) experts (edited by Marty)
  - Spring, Hibernate, EJB3, Ruby/Rails

## Tags Developed

(Two versions of each: Java-based and tag-file-based)

- **contextPath**
  - Outputs the Web application context path (e.g., /myApp), to simplify relative URLs.
- **simpleAlert**
  - Takes the result of a URL and puts it in popup dialog box.
- **alert**
  - Takes the result of a URL and puts it in popup dialog box. Sends data from designated input element.
- **simpleButton**
  - Takes the result of a URL and puts it in designated HTML element.
- **button**
  - Takes the result of a URL and puts it in specified HTML element. Sends data from list of designated input elements.

5

J2EE training: <http://courses.coreservlets.com>

## Review: Java-Based Tags

- **Tag handler class**
  - Extend SimpleTagSupport
  - Override doTag
  - Get the JspWriter with getJspContext().getOut()
  - Use the JspWriter to generate output
  - Define setBlah for each attribute named blah
- **TLD File**
  - /WEB-INF/somewhere/somename.tld
  - <uri>http://fake-address</uri>
  - <tag> ... </tag>
    - Contains description (optional), name, tag-class, body-content, attribute (one for each attribute)
- **JSP File**
  - <% @ taglib uri="http://fake-address" prefix="blah" %>
  - <blah:tagName/> or <blah:tagName>...</blah:tagName>

6

J2EE training: <http://courses.coreservlets.com>

# Review: Tag Files

- **Tag File**

- /WEB-INF/tags/tagName.tag
- Create chunk of JSP that generates the output
- Declare attributes with `<% @ attribute ...%>`
- Output attributes with `${attributeName}`

- **JSP File**

- `<% @ taglib tagdir="/WEB-INF/tags prefix="blah" %>`
- `<blah:tagName/>` or `<blah:tagName>...</blah:tagName>`

7

J2EE training: <http://courses.coreservlets.com>

# contextPath Tag

- **Goal**

- Make a tag that outputs the Web application context path.
- This will simplify relative URLs for loading style sheets and JavaScript files.

- **Sample Usage**

```
<link rel="stylesheet"
      href="<ajax:contextPath/>/css/styles.css"
      type="text/css"/>
<script src="<ajax:contextPath/>/scripts/ajax.js"
        type="text/javascript"></script>
```

8

J2EE training: <http://courses.coreservlets.com>

## contextPath (Java-Based): Tag Handler

```
package coreservlets.tags;

import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
import java.io.*;
import javax.servlet.http.*;

public class ContextPathTag extends SimpleTagSupport {
    public void doTag() throws JspException, IOException {
        PageContext pageContext = (PageContext)getJspContext();
        HttpServletRequest request =
            (HttpServletRequest)pageContext.getRequest();
        JspWriter out = getJspContext().getOut();
        out.print(request.getContextPath());
    }
}
```

9

J2EE training: <http://courses.coreservlets.com>

## contextPath (Java-Based): TLD File

- In /WEB-INF/tlds/ajax-tags.tld

```
<?xml version="1.0" encoding="UTF-8" ?>
<taglib ...
    version="2.0">
  <description>Some simple Ajax tags.</description>
  <tlib-version>1.0</tlib-version>
  <short-name>simple-ajax-tags</short-name>
  <uri>http://coreservlets.com/ajax-tags</uri>

  <tag>
    <description>Context path of the Web app</description>
    <name>contextPath</name>
    <tag-class>coreservlets.tags.ContextPathTag</tag-class>
    <body-content>empty</body-content>
  </tag>
```

10

J2EE training: <http://courses.coreservlets.com>

## contextPath (Java-Based): JSP Page

```
<!DOCTYPE ...>
<%@ taglib uri="http://coreservlets.com/ajax-tags"
    prefix="ajax" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Ajax: Custom Tags (Java Version)</title>
<link rel="stylesheet"
    href="<ajax:contextPath/>/css/styles.css"
    type="text/css"/>
<script src="<ajax:contextPath/>/scripts/ajax.js"
    type="text/javascript"></script>
</head>
...
```

## contextPath (Tag-File-Based): Tag File

- **/WEB-INF/tags/contextPath.tag**

```
<%= request.getContextPath() %>
```

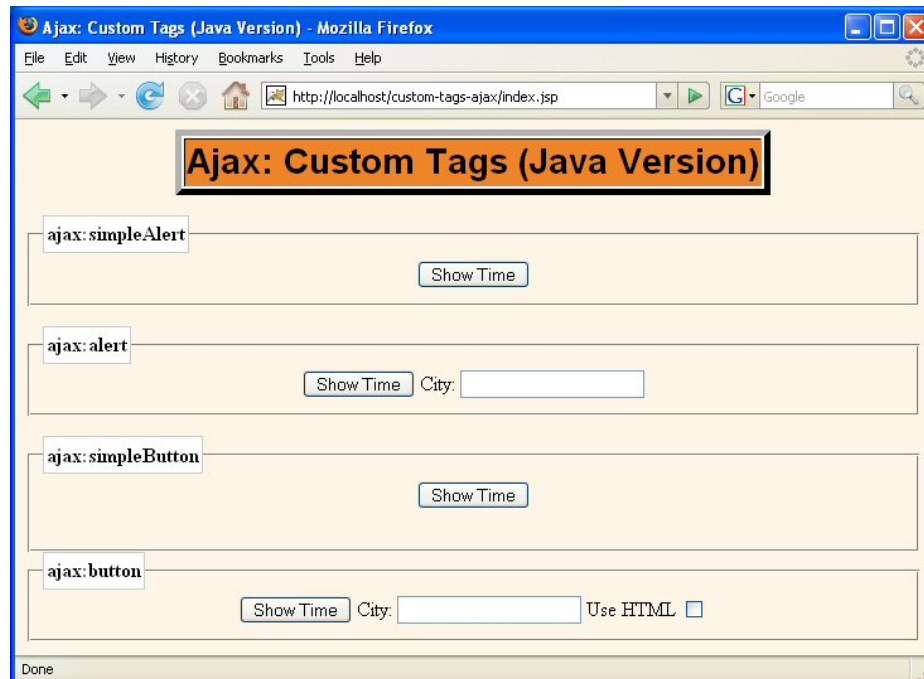
# contextPath (Tag-File-Based): JSP File

```
<!DOCTYPE ...>
<%@ taglib tagdir="/WEB-INF/tags"
      prefix="ajax" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Ajax: Custom Tags (Tag File Version)</title>
<link rel="stylesheet"
      href="<ajax:contextPath/>/css/styles.css"
      type="text/css"/>
<script src="<ajax:contextPath/>/scripts/ajax.js"
        type="text/javascript"></script>
</head>
```

13

J2EE training: <http://courses.coreservlets.com>

# contextPath: Result



14

J2EE training: <http://courses.coreservlets.com>

# simpleAlert Tag

- **Goal**

- Make a tag that generates a button.
- Clicking on the button will result in a designated URL being contacted, and the result of that URL being put into a popup dialog box.

- **Sample Usage**

```
<fieldset>
  <legend>ajax:simpleAlert</legend>
  <ajax:simpleAlert address="show-time"
                    label="Show Time"/>
</fieldset>
```

# Supporting JavaScript for All Ajax Tags

```
var request;

function getRequestObject() {
  if (window.ActiveXObject) {
    return(new ActiveXObject("Microsoft.XMLHTTP"));
  } else if (window.XMLHttpRequest) {
    return(new XMLHttpRequest());
  } else {
    return(null);
  }
}
```

## simpleAlert: Supporting JavaScript

```
function simpleAjaxAlert(address) {
    request = getRequestObject();
    var responseHandler = showAlert;
    request.onreadystatechange = responseHandler;
    request.open("POST", address, true);
    request.setRequestHeader("Content-Type",
                            "application/x-www-form-urlencoded");
    request.send(null);
}

function showAlert() {
    if ((request.readyState == 4) &&
        (request.status == 200)) {
        alert(request.responseText);
    }
}
```

17

J2EE training: <http://courses.coreservlets.com>

## simpleAlert (Java-Based): Tag Handler

```
public class SimpleAlertTag extends SimpleTagSupport {
    private String address, label;

    public void setAddress(String address) {
        this.address = address;
    }

    public void setLabel(String label) {
        this.label = label;
    }

    public void doTag() throws JspException, IOException {
        JspWriter out = getJspContext().getOut();
        String output =
            String.format("<input type=\"button\" value=\"%s\" " +
                          "onclick='simpleAjaxAlert(\"%s\")' />",
                          label, address);
        out.print(output);
    }
}
```

18

J2EE training: <http://courses.coreservlets.com>



## simpleAlert (Java-Based): TLD File

```
<?xml version="1.0" encoding="UTF-8" ?>
<taglib ... version="2.0"> ...
  <uri>http://coreservlets.com/ajax-tags</uri>
  ...
  <tag>
    <description>
      A button that triggers a simple Ajax alert
    </description>
    <name>simpleAlert</name>
    <tag-class>coreservlets.tags.SimpleAlertTag</tag-class>
    <body-content>empty</body-content>
    <attribute>
      <name>address</name>
      <required>>true</required>
    </attribute>
    <attribute>
      <name>label</name>
      <required>>true</required>
    </attribute>
  </tag>
```

19

J2EE training: <http://courses.coreservlets.com>

## simpleAlert (Java-Based): JSP Page

```
<!DOCTYPE ...>
<%@ taglib uri="http://coreservlets.com/ajax-tags"
  prefix="ajax" %>
<html xmlns="http://www.w3.org/1999/xhtml">
  ...
  <body>
    ...
    <form action="#">
      <fieldset>
        <legend>ajax:simpleAlert</legend>
        <ajax:simpleAlert address="show-time"
          label="Show Time"/>
      </fieldset>
```

20

J2EE training: <http://courses.coreservlets.com>

## simpleAlert (Tag-File-Based): Tag File

- /WEB-INF/tags/simpleAlert.tag

```
<%@ attribute name="address" required="true" %>
<%@ attribute name="label" required="true" %>
<input type="button" value="{label}"
      onclick='simpleAjaxAlert("{address}")' />
```

## simpleAlert (Tag-File-Based): JSP File

```
<!DOCTYPE ...>
<%@ taglib tagdir="/WEB-INF/tags"
      prefix="ajax" %>
<html xmlns="http://www.w3.org/1999/xhtml">
...
<body>
...
<form action="#">
<fieldset>
  <legend>ajax:simpleAlert</legend>
  <ajax:simpleAlert address="show-time"
                    label="Show Time"/>
</fieldset>
```

## simpleAlert: Server-Side Code (Servlet)

```
public class ShowTime extends HttpServlet {
    public void doPost(HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException {
        response.setHeader("Cache-Control", "no-cache");
        response.setHeader("Pragma", "no-cache");
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Date currentTime = new Date();
        String message =
            String.format("It is now %tr on %tD.",
                          currentTime, currentTime);
        out.print(message);
    }
}
```

23

J2EE training: <http://courses.coreservlets.com>

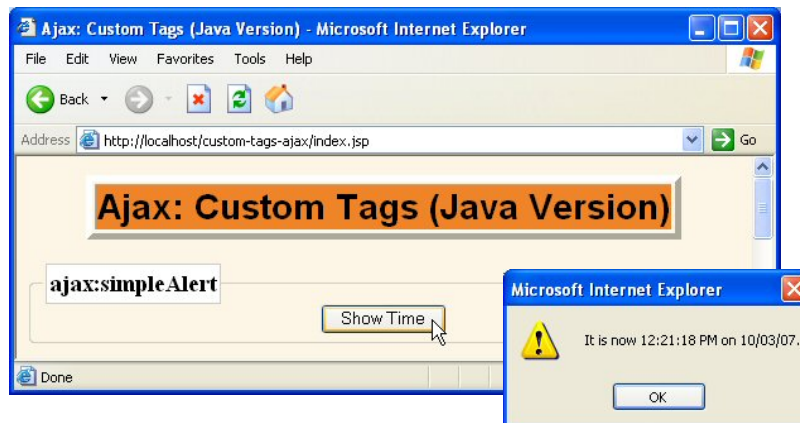
## simpleAlert: Server-Side Code (web.xml)

```
...
<servlet>
    <servlet-name>ShowTime</servlet-name>
    <servlet-class>coreservlets.ShowTime</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>ShowTime</servlet-name>
    <url-pattern>/show-time</url-pattern>
</servlet-mapping>
...
```

24

J2EE training: <http://courses.coreservlets.com>

# simpleAlert: Results



25

J2EE training: <http://courses.coreservlets.com>

# alert Tag

- **Goal**
  - Similar to simpleAlert: builds a button that contacts URL and puts result in popup dialog box
  - Data of the form fieldName=fieldValue sent for the designated input element

- **Sample Usage**

```
<fieldset>
  <legend>ajax:alert</legend>
  <ajax:alert address="show-time-in-city"
    label="Show Time"
    field="city1"/>
  City:&nbsp;  <input type="text" name="city" id="city1"/>
</fieldset>
```

26

J2EE training: <http://courses.coreservlets.com>

## alert: Supporting JavaScript (showAlert Function Shown Earlier)

```
function ajaxAlert(address, fieldID) {
    request = getRequestObject();
    var responseHandler = showAlert;
    request.onreadystatechange = responseHandler;
    request.open("POST", address, true);
    request.setRequestHeader("Content-Type",
        "application/x-www-form-urlencoded");
    var fieldName = getName(fieldID);
    var data = fieldName + "=" +
        escape(getValue(fieldID));
    request.send(data);
}

function getName(id) {
    return(getElement(id).name);
}

function getElement(id) {
    return(document.getElementById(id));
}
```

27

J2EE training: <http://courses.coreservlets.com>

## alert (Java-Based): Tag Handler

```
public class AlertTag extends SimpleTagSupport {
    private String address, label, field;

    public void setAddress(String address) {
        this.address = address;
    }
    public void setLabel(String label) {
        this.label = label;
    }
    public void setField(String field) {
        this.field = field;
    }

    public void doTag() throws JspException, IOException {
        JspWriter out = getJspContext().getOut();
        String output =
            String.format("<input type=\"button\" value=\"%s\" " +
                "onclick='ajaxAlert(\"%s\", \"%s\")'/>",
                label, address, field);
        out.print(output);
    }
}
```

28

J2EE training: <http://courses.coreservlets.com>

## alert (Java-Based): TLD File

```
<tag>
  <description>
    A button that triggers an Ajax alert
  </description>
  <name>alert</name>
  <tag-class>coreservlets.tags.AlertTag</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>address</name>
    <required>>true</required>
  </attribute>
  <attribute>
    <name>label</name>
    <required>>true</required>
  </attribute>
  <attribute>
    <name>field</name>
    <required>>true</required>
  </attribute>
</tag>
```

29

J2EE training: <http://courses.coreservlets.com>

## alert (Java-Based): JSP Page

```
<!DOCTYPE ...>
<%@ taglib uri="http://coreservlets.com/ajax-tags"
    prefix="ajax" %>
<html xmlns="http://www.w3.org/1999/xhtml">
  ...
  <form action="#">
    ...
    <fieldset>
      <legend>ajax:alert</legend>
      <ajax:alert address="show-time-in-city"
        label="Show Time"
        field="city1"/>
      City:&nbsp;  <input type="text" name="city" id="city1"/>
    </fieldset>
```

30

J2EE training: <http://courses.coreservlets.com>

## alert (Tag-File-Based): Tag File

- /WEB-INF/tags/alert.tag

```
<%@ attribute name="address" required="true" %>
<%@ attribute name="label" required="true" %>
<%@ attribute name="field" required="true" %>
<input type="button" value="{label}"
      onclick='ajaxAlert("{address}", "{field}")' />
```

## alert (Tag-File-Based): JSP File

```
<!DOCTYPE ...>
<%@ taglib tagdir="/WEB-INF/tags"
      prefix="ajax" %>
<html xmlns="http://www.w3.org/1999/xhtml">
...
<form action="#">
...
<fieldset>
  <legend>ajax:alert</legend>
  <ajax:alert address="show-time-in-city"
            label="Show Time"
            field="city1"/>
  City:&nbsp;  <input type="text" name="city" id="city1"/>
</fieldset>
```

## alert: Server-Side Code (Servlet)

```
public class ShowTimeInCity extends HttpServlet {
    public void doPost(HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException {
        response.setHeader("Cache-Control", "no-cache");
        response.setHeader("Pragma", "no-cache");
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String city = request.getParameter("city");
        boolean htmlFlag = true;
        String htmlParam = request.getParameter("useHTML");
        if ((htmlParam == null) || (htmlParam.trim().equals(""))) {
            htmlFlag = false;
        }
        String message = TimeZone.getTimeString(city);
        if (htmlFlag) {
            message = String.format("<h2>%s</h2>", message);
        }
        out.print(message);
    }
}
```

33

J2EE training: <http://courses.coreservlets.com>

## alert: Server-Side Code (Supporting Class)

```
public class TimeZone {

    // Details omitted

    public static String getTimeString(String city,
                                       TimeZone[] zones,
                                       Calendar dateTime) {
        int timeZoneOffset = findOffset(city);
        if (timeZoneOffset <= 0) {
            dateTime.add(Calendar.HOUR_OF_DAY, timeZoneOffset);
            String timeString =
                String.format("In %s, it is now %tr on %tD.",
                              city, dateTime, dateTime);
            return(timeString);
        } else {
            String timeString =
                String.format("Unknown city '%s'.", city);
            return(timeString);
        }
    }
}
```

34

J2EE training: <http://courses.coreservlets.com>



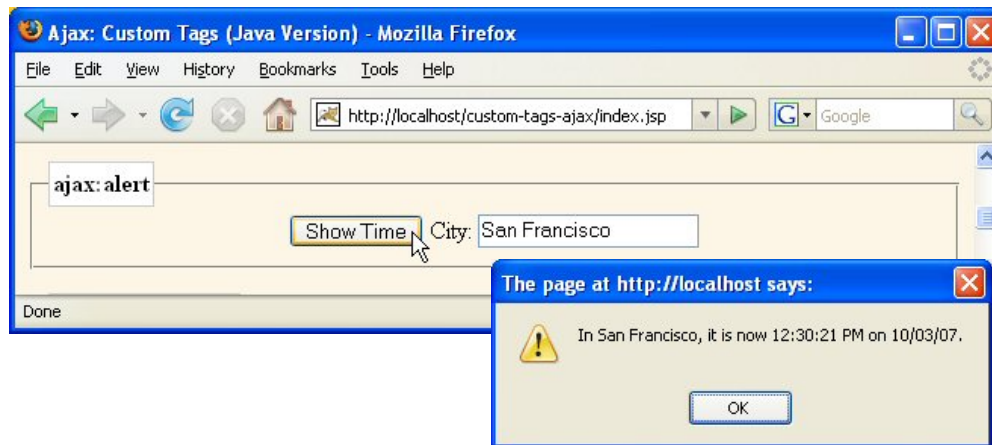
## alert: Server-Side Code (web.xml)

```
...  
<servlet>  
  <servlet-name>ShowTimeInCity</servlet-name>  
  <servlet-class>  
    coreservlets.ShowTimeInCity  
  </servlet-class>  
</servlet>  
<servlet-mapping>  
  <servlet-name>ShowTimeInCity</servlet-name>  
  <url-pattern>/show-time-in-city</url-pattern>  
</servlet-mapping>  
...
```

35

J2EE training: <http://courses.coreservlets.com>

## alert: Results



36

J2EE training: <http://courses.coreservlets.com>

# simpleButton Tag

- **Goal**

- Make a tag that generates a button.
- Clicking on the button will result in a designated URL being contacted, and the result of that URL being put into the HTML region that has designated id.

- **Sample Usage**

```
<fieldset>
  <legend>ajax:simpleButton</legend>
  <ajax:simpleButton address="show-time"
                    label="Show Time"
                    resultRegion="result1"/>
  <h2 id="result1"></h2>
</fieldset>
```

37

J2EE training: <http://courses.coreservlets.com>

# simpleButton: Supporting JavaScript

```
function simpleAjaxRequest(address, region) {
  request = getRequestObject();
  var responseHandler =
    function() { showResponseText(region); };
  request.onreadystatechange = responseHandler;
  request.open("POST", address, true);
  request.setRequestHeader("Content-Type",
                          "application/x-www-form-urlencoded");
  request.send(null);
}

function showResponseText(region) {
  if ((request.readyState == 4) &&
      (request.status == 200)) {
    document.getElementById(region).innerHTML =
      request.responseText;
  }
}
```

38

J2EE training: <http://courses.coreservlets.com>

## simpleButton (Java-Based): Tag Handler

```
public class SimpleButtonTag extends SimpleTagSupport {
    private String address, label, resultRegion;

    public void setAddress(String address) {
        this.address = address;
    }
    public void setLabel(String label) {
        this.label = label;
    }
    public void setResultRegion(String resultRegion) {
        this.resultRegion = resultRegion;
    }

    public void doTag() throws JspException, IOException {
        JspWriter out = getJspContext().getOut();
        String output =
            String.format("<input type=\"button\" value=\"%s\" " +
                "onclick='simpleAjaxRequest(\"%s\", \"%s\")'/>",
                label, address, resultRegion);
        out.print(output);
    }
}
```

39

J2EE training: <http://courses.coreservlets.com>

## simpleButton (Java-Based): TLD File

```
<tag>
  <description>
    A button that triggers a simple Ajax request
  </description>
  <name>simpleButton</name>
  <tag-class>coreservlets.tags.SimpleButtonTag</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>address</name>
    <required>true</required>
  </attribute>
  <attribute>
    <name>label</name>
    <required>true</required>
  </attribute>
  <attribute>
    <name>resultRegion</name>
    <required>true</required>
  </attribute>
</tag>
```

40

J2EE training: <http://courses.coreservlets.com>

## simpleButton (Java-Based): JSP Page

```
<!DOCTYPE ...>
<%@ taglib uri="http://coreservlets.com/ajax-tags"
      prefix="ajax" %>
<html xmlns="http://www.w3.org/1999/xhtml">
  ...
  <form action="#">
    ...
    <fieldset>
      <legend>ajax:simpleButton</legend>
      <ajax:simpleButton address="show-time"
        label="Show Time"
        resultRegion="result1"/>
      <h2 id="result1"></h2>
    </fieldset>
```

## simpleButton (Tag-File-Based): Tag File

- /WEB-INF/tags/simpleButton.tag

```
<%@ attribute name="address" required="true" %>
<%@ attribute name="label" required="true" %>
<%@ attribute name="resultRegion" required="true" %>
<input type="button" value="{label}"
  onclick=
    'simpleAjaxRequest("{address}", "{resultRegion}")' />
```

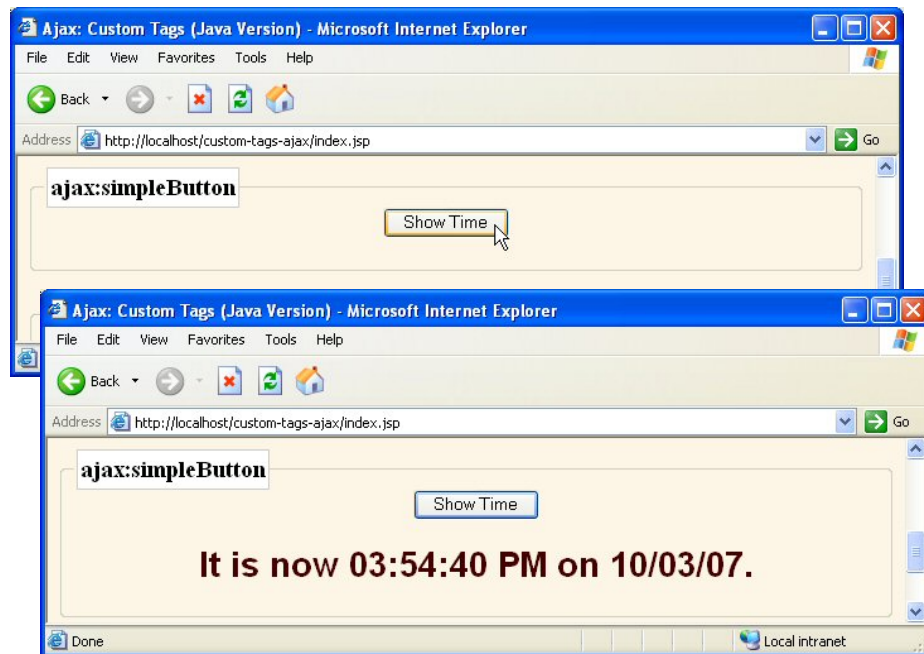
## simpleButton (Tag-File-Based): JSP File

```
<!DOCTYPE ...>
<%@ taglib tagdir="/WEB-INF/tags"
      prefix="ajax" %>
<html xmlns="http://www.w3.org/1999/xhtml">
  ...
  <form action="#">
    ...
    <fieldset>
      <legend>ajax:simpleButton</legend>
      <ajax:simpleButton address="show-time"
                        label="Show Time"
                        resultRegion="result1"/>
      <h2 id="result1"></h2>
    </fieldset>
```

## simpleButton: Server-Side Code

- **Servlet**
  - ShowTime servlet shown earlier in simpleAlert example
- **web.xml**
  - Servlet-mapping of show-time shown earlier in simpleAlert example

# simpleButton: Results



45

J2EE training: <http://courses.coreservlets.com>

# button Tag

- **Goal**
  - Similar to simpleButton: builds a button that contacts URL and puts result in designated HTML region
  - Takes a list of field names and sends data of the form `fieldName1=fieldValue1&fieldName2=fieldValue2&...&fieldNameN=fieldValueN`

- **Sample Usage**

```
<fieldset>
  <legend>ajax:button</legend>
  <ajax:button address="show-time-in-city"
    label="Show Time"
    resultRegion="result2"
    fields="city2,useHTML" />
  City:&nbsp;<input type="text" name="city" id="city2" />
  Use HTML:&nbsp;<input type="checkbox" name="useHTML"
    id="useHTML" /><br />
  <div id="result2"></div>
</fieldset>
```

46

J2EE training: <http://courses.coreservlets.com>

## button: Supporting JavaScript (showResponseText Shown Earlier)

```
function ajaxRequest(address, region /* ... */) { // Varargs!
    request = getRequestObject();
    var responseHandler =
        function() { showResponseText(region); };
    request.onreadystatechange = responseHandler;
    request.open("POST", address, true);
    request.setRequestHeader("Content-Type",
                            "application/x-www-form-urlencoded");

    var data = "";
    for(i=2; i<arguments.length; i++) {
        var fieldID = arguments[i];
        var fieldName = getName(fieldID);
        data = data + fieldName + "=" +
            escape(getValue(fieldID)) + "&";
    }
    request.send(data);
}
```

## button: Supporting JavaScript Continued (getElement Shown Earlier)

```
// For radio buttons and checkboxes, their value is "on" if they
// are checked, an empty string otherwise. For textboxes and
// most other elements, their value is the "value" property.

function getValue(id) {
    var element = getElement(id);
    var value;
    if ((element.type == "checkbox") || (element.type=="radio")) {
        if (element.checked == true) {
            return("on");
        } else {
            return("");
        }
    }
    return(element.value);
}
```

## button (Java-Based): Tag Handler

```
public class ButtonTag extends SimpleTagSupport {
    private String address, label, resultRegion, fields;

    public void setAddress(String address) {
        this.address = address;
    }

    // Also setLabel, setResultRegion, and setFields

    public void doTag() throws JspException, IOException {
        String argList = AjaxUtils.buildArgList(fields);
        JspWriter out = getJspContext().getOut();
        String output =
            String.format("<input type=\"button\" value=\"%s\" " +
                "onclick='ajaxRequest(\"%s\", \"%s\", %s)'/>",
                label, address, resultRegion, argList);
        out.print(output);
    }
}
```

49

J2EE training: <http://courses.coreservlets.com>

## button (Java-Based): TLD File

```
<tag>
  <description>Button that triggers Ajax request</description>
  <name>button</name>
  <tag-class>coreservlets.tags.ButtonTag</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>address</name>
    <required>>true</required>
  </attribute>
  <attribute>
    <name>label</name>
    <required>>true</required>
  </attribute>
  <attribute>
    <name>resultRegion</name>
    <required>>true</required>
  </attribute>
  <attribute>
    <name>fields</name>
    <required>>true</required>
  </attribute>
</tag>
```

50

J2EE training: <http://courses.coreservlets.com>



## button (Java-Based): JSP Page

```
<!DOCTYPE ...>
<%@ taglib uri="http://coreservlets.com/ajax-tags"
      prefix="ajax" %>
<html xmlns="http://www.w3.org/1999/xhtml">...
<form action="#">
...
<fieldset>
  <legend>ajax:button</legend>
  <ajax:button address="show-time-in-city"
              label="Show Time"
              resultRegion="result2"
              fields="city2,useHTML"/>
  City:&nbsp;<input type="text" name="city" id="city2"/>
  Use HTML&nbsp;<input type="checkbox" name="useHTML"
                  id="useHTML"/><br/>
  <div id="result2"></div>
</fieldset>
```

51

J2EE training: <http://courses.coreservlets.com>

## button (Tag-File-Based): Tag File

- /WEB-INF/tags/alert.tag

```
<%@ attribute name="address" required="true" %>
<%@ attribute name="label" required="true" %>
<%@ attribute name="field" required="true" %>
<input type="button" value="${label}"
      onclick='ajaxAlert("${address}", "${field}")' />
```

52

J2EE training: <http://courses.coreservlets.com>

## button (Tag-File-Based): JSP File

```
<!DOCTYPE ...>
<%@ taglib tagdir="/WEB-INF/tags"
      prefix="ajax" %>
<html xmlns="http://www.w3.org/1999/xhtml">...
<form action="#">
...
<fieldset>
  <legend>ajax:button</legend>
  <ajax:button address="show-time-in-city"
              label="Show Time"
              resultRegion="result2"
              fields="city2,useHTML"/>
  City:&nbsp;<input type="text" name="city" id="city2"/>
  Use HTML&nbsp;<input type="checkbox" name="useHTML"
                 id="useHTML"/><br/>
  <div id="result2"></div>
</fieldset>
```

53

J2EE training: <http://courses.coreservlets.com>

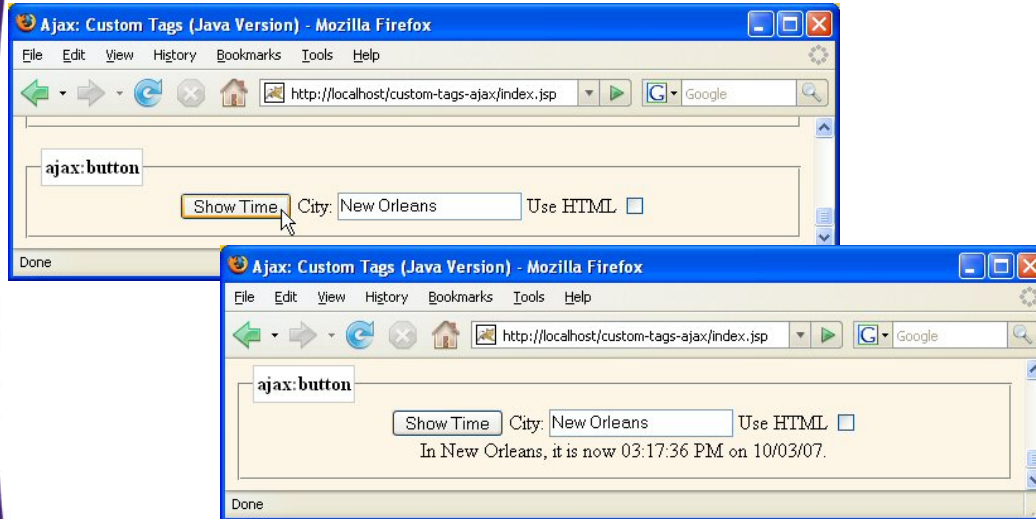
## button: Server-Side Code

- **Servlet**
  - ShowTimeInCity servlet shown earlier in alert example
- **web.xml**
  - Servlet-mapping of show-time-in-city shown earlier in alert example

54

J2EE training: <http://courses.coreservlets.com>

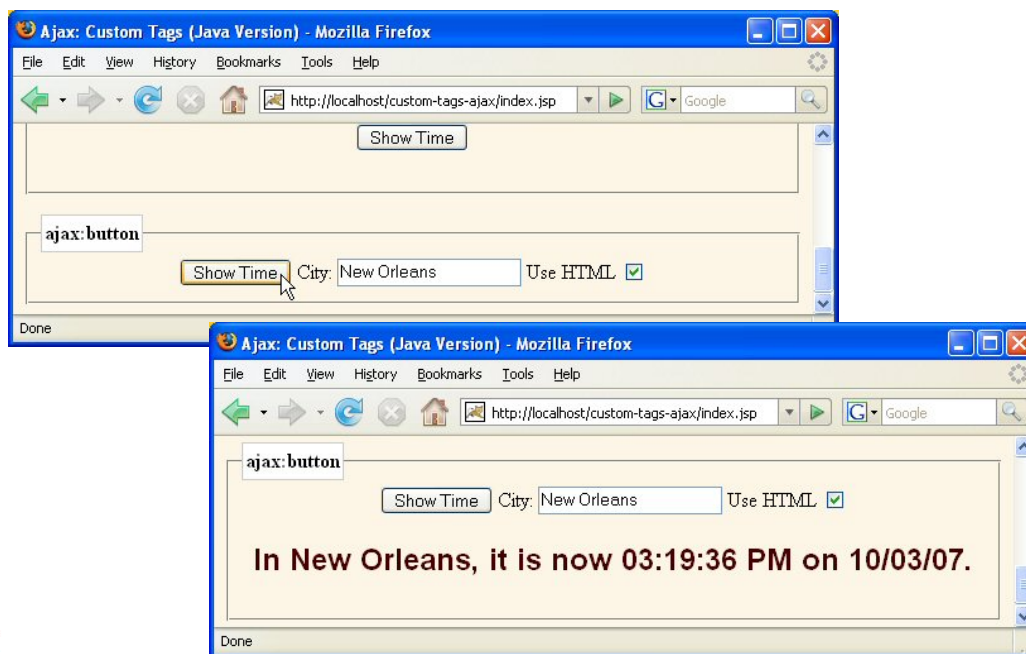
# button: Results



55

J2EE training: <http://courses.coreservlets.com>

# button: Results Continued



56

J2EE training: <http://courses.coreservlets.com>

# Summary

- **It is straightforward to build tags to support some of the most common Ajax functionality**
  - Doing so greatly simplifies work for the page developer
- **The tag file approach tends to be superior to the Java-based approach**
  - Because the hard work is done in the server-side code and in the JavaScript, so the tags are mostly a matter of formatting
  - When Java-based tags are used, remember `String.format`

57

J2EE training: <http://courses.coreservlets.com>

© 2007 Marty Hall



# Questions?

**Customized J2EE Training: <http://courses.coreservlets.com/>**

Servlets, JSP, Struts, JSF/MyFaces, Hibernate, Ajax, Java 5, Java 6, etc. Ruby/Rails coming soon.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

58