



JSF: The Ajax4jsf Library

Originals of Slides and Source Code for Examples:
<http://www.coreservlets.com/JSF-Tutorial/>

Customized J2EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces, Hibernate, Ajax, GWT, 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 JSF training, please 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 courses.
- Courses developed and taught by coreservlets.com experts (edited by Marty)
 - Spring, Hibernate, EJB3, Ruby/Rails

Contact hall@coreservlets.com for details

Topics in This Section

- **Ajax motivation**
- **Installation**
- **Main Ajax4jsf Elements**
 - a4j:commandButton
 - a4j:commandLink
 - a4j:poll
 - a4j:support
- **Limitations on use of h:outputText with Ajax4jsf**
- **Other Ajax Toolkits**

5

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

© 2007 Marty Hall



Motivation

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

Servlets, JSP, Struts, JSF/MyFaces, Hibernate, Ajax, GWT, 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.



Installation and Setup

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

Servlets, JSP, Struts, JSF/MyFaces, Hibernate, Ajax, GWT, 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.

Installing Ajax4jsf

- **Download latest binary version**
 - <http://labs.jboss.com/jbossajax4jsf/downloads/>
- **Unzip**
 - Into any location
- **Install ajax4jsf.jar**
 - Copy /lib/ajax4jsf.jar into the WEB-INF/lib directory of your Web app
- **Add filter to web.xml**
 - See next page
- **Shortcut**
 - Download ajax4jsf-blank from <http://www.coreservlets.com/JSF-Tutorial/>
 - Web app with all MyFaces and Ajax4jsf JAR files and web.xml entries

Filter Settings for web.xml

```
<?xml version="1.0"?>
<web-app ...>
  ...
  <filter>
    <display-name>Ajax4jsf Filter</display-name>
    <filter-name>ajax4jsf</filter-name>
    <filter-class>org.ajax4jsf.Filter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>ajax4jsf</filter-name>
    <servlet-name>Faces Servlet</servlet-name>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>INCLUDE</dispatcher>
  </filter-mapping>
</web-app>
```

11

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

Using Ajax4jsf

- **Use xhtml**
 - Most Ajax applications use xhtml, not HTML 4
- **Add taglib for ajax4jsf**
 - `<% @ taglib uri="https://ajax4jsf.dev.java.net/ajax" prefix="a4j"% >`
- **Give ids to sections that you want to update**
 - `<h:outputText ... id="someName"/>`
 - Note: ids within a page must be unique
- **Use a4j: tags**
 - Almost all *must* go inside h:form
 - Even the ones that don't use form elements (e.g., a4j:poll)

12

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

Basic Template

```
<?xml version="1.0" encoding="UTF-8"?>
<%@ taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<%@ taglib uri="https://ajax4jsf.dev.java.net/ajax"
    prefix="a4j"%>
<f:view>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<meta http-equiv="content-type"
    content="text/html; charset=UTF-8" />
<head><title>Some Title</title>
</head>
<body>
...
</body></html>
</f:view>
```

XML header recommended by xhtml specification, but you should omit because of Internet Explorer bug.

13

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

XHTML: Case

- In HTML 4, case does not matter for tag names and attribute names
 - <BODY>, <Body>, and <body> are equivalent
 - <H1 ALIGN="..."> is equivalent to <H1 aLiGn="...">
- In xhtml, tag names and attribute names must be in lower case
 - <body>, <h1 align="...">

14

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

XHTML: Quotes

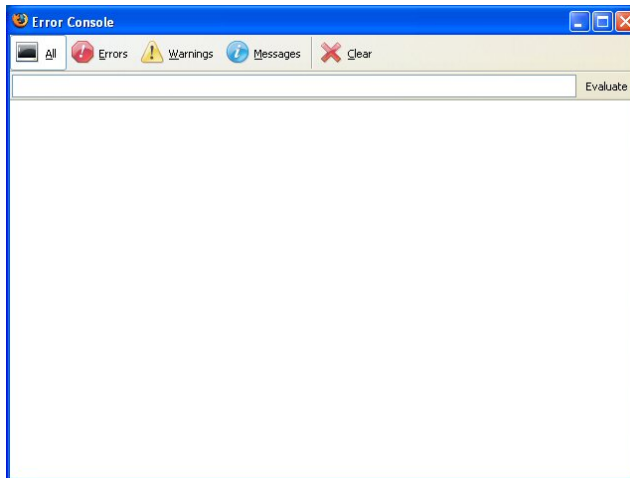
- In HTML 4, quotes are optional if attribute value contains only alphanumeric values
 - `<H1 ALIGN="LEFT">` or `<H1 ALIGN=LEFT>`
- In xhtml, you must always use single or double quotes
 - `<h1 align="left">` or `<h1 align='left'>`

XHTML: End Tags

- **HTML 4**
 - Some tags are containers
 - `<H1>...</H1>`, `<A HREF...>...`
 - Some tags are standalone
 - `
`, `<HR>`
 - Some tags have optional end tags
 - `<P>`, ``, `<TR>`, `<TD>`, `<TH>`
- **XHTML**
 - All tags are containers. End tags always required.
 - `<p>...</p>`, `...`
 - If there is no body content, start/end tags can be merged
 - `
</br>` or just `
`

Ajax and the Firefox JavaScript Console

- Invoke with Control-Shift-J



- Also see Venkman JavaScript debugger
 - <http://www.mozilla.org/projects/venkman/>
 - <https://addons.mozilla.org/firefox/216/>

17

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

© 2007 Marty Hall



Main Ajax4jsf Tags

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

Servlets, JSP, Struts, JSF/MyFaces, Hibernate, Ajax, GWT, 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.

Tag Summary

- **a4j:commandButton** and **a4j:commandLink**
 - Run code on the server, then update specified JSF element (or comma separated element list) after.
 - `<a4j:commandButton action="#{bean.method}" value="Button label" reRender="some-id"/>`
... `<h:outputText value="#{bean.prop}" id="some-id"/>`
- **a4j:poll**
 - Run code periodically on server, then update specified JSF element(s)
- **a4j:support**
 - Capture JavaScript event in *any* existing JSF control and invoke server-side code, then update specified element(s)
 - `<h:inputText ...>`
 `<a4j:support event="onkeyup" reRender="some-id"/>`
 `</h:inputText>`

19

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

a4j:commandButton: Basic Syntax

```
<a4j:commandButton  
  action="#{bean.method}"  
  value="Button label"  
  reRender="some-id"/>
```

When you press the button, send a behind-the-scenes, asynchronous HTTP call to server, and run this method. This method should look like a normal action controller (i.e., no arguments and returns a String), but the return value is ignored (i.e., is not used to match a navigation rule in faces config).

After action method is executed, re-evaluate and re-display the JSF element that has this id. The point is that this JSF element should output something that changed as a result of the call to the action controller.

20

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

Reminder: Template

```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="https://ajax4jsf.dev.java.net/ajax" prefix="a4j"%>
<f:view>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet"
  href="/css/styles.css"
  type="text/css"/>
<title>Ajax4JSF Examples</title>
</head>
<body> ...
```

21

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

h:commandButton: Example

```
<tr><td width="100">
  <h:form>
    <a4j:commandButton
      action="#{numBean.makeResult}"
      value="Show Random Number"
      reRender="region1" />
  </h:form>
</td>
<td width="100">
  <h:outputText value="#{numBean.result}"
    id="region1" />
</td>
</tr>
```

22

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

h:commandLink: Example

```
<tr><td width="100">
  <h:form>
    <a4j:commandLink
      action="#{numBean.makeResult}"
      value="Show Random Number"
      reRender="region2"/>
  </h:form>
</td>
<td width="100">
  <h:outputText value="#{numBean.result}"
    id="region2"/>
</td>
</tr>
```

Remember that ids must be unique within a page.

23

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

a4j:commandButton and a4j:commandLink: Bean

```
public class RandomNumberBean {
  private int range = 1;
  private double result;

  public int getRange() {
    return(range);
  }

  public void setRange(int range) {
    this.range = range;
  }

  public String makeResult() {
    result = Math.random() * range;
    return(null);
  }

  public double getResult() {
    return(result);
  }
}
```

Action controllers must have String return type to satisfy JSF API.

However, return values are ignored, since navigation rules do not apply.

24

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

a4j:commandButton and a4j:commandLink: faces-config

```
<managed-bean>
  <managed-bean-name>
    numBean
  </managed-bean-name>
  <managed-bean-class>
    coreservlets.RandomNumberBean
  </managed-bean-class>
  <managed-bean-scope>
    request
  </managed-bean-scope>
</managed-bean>
```

a4j:commandButton: Example 2

- **a4j:commandButton results in normal JSF request processing cycle**
 - So, form is still submitted
 - Bean setter methods still execute
 - Action controller runs *after* setter methods, so controller has access to results of all form fields

a4j:commandButton: Example 2

```
<tr><td width="100">
    <h:form>
        Range:
        <h:inputText value="#{numBean.range}"
                    size="5"/><br/>
        <a4j:commandButton
            action="#{numBean.makeResult}"
            value="Show Random Number"
            reRender="region3"/>
    </h:form>
</td>
<td width="100">
    <h:outputText value="#{numBean.result}"
                  id="region3"/>
</td>
</tr>
```

27

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

a4j:commandButton and a4j:commandLink: Bean

```
public class RandomNumberBean {
    private int range = 1;
    private double result;

    public int getRange() {
        return(range);
    }

    public void setRange(int range) {
        this.range = range;
    }

    public String makeResult() {
        result = Math.random() * range;
        return(null);
    }

    public double getResult() {
        return(result);
    }
}
```

28

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

a4j:commandButton and a4j:commandLink: Results

The image displays two screenshots of a Mozilla Firefox browser window titled "Ajax4jsf Examples". The browser address bar shows "http://localhost/ajax4jsf/welcome.faces". The page content includes a header "Ajax4jsf Examples" and a section titled "a4j:commandButton and a4j:commandLink".

The left screenshot shows a table with two columns: "Control" and "Result". The table contains three rows of controls and their results:

Control	Result
Show Random Number	0.0
Show Random Number	0.0
Range: 1 Show Random Number	0.0

The right screenshot shows the same table after the controls have been interacted with. The results are now random numbers:

Control	Result
Show Random Number	0.8080057864819878
Show Random Number	0.5656789493724107
Range: 150 Show Random Number	129.31059414654032

Limitations on Use of h:outputText with Ajax4jsf

- **In JSF, the following is perfectly legal**
 - `<body bgcolor=`
`"<h:outputText #{myBean.fgColor}"/>">`
 - Results in `<body bgcolor="red">`
- **But the following is illegal**
 - `<body bgcolor=`
`"<h:outputText #{myBean.fgColor}" id="foo"/>">`
 - Results in `<body bgcolor="red">`
- **In regular JSF, this is no problem**
 - Since there is no need for ids
- **But with Ajax, you need a DOM-accessible region to update**
 - So, you are limited in the places that you can update asynchronously
 - You can only update separate xhtml elements, not just sections

a4j:poll: Basic Syntax

```
<a4j:poll  
  interval="x"  
  reRender="some-id"/>
```

- **Interpretation**

- Every x milliseconds, send off an asynchronous HTTP request to the server.
- Re-evaluate and re-display the JSF element with the id *some-id*.
 - Key point: this element should result from code that gives different values at different times
 - Unlike with `a4j:commandButton`, there is no explicit additional server-side method to run

31

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

a4j:poll: Example

```
<h:form>  
  <a4j:poll interval="5000"  
    reRender="timeDisplay"/>  
  <h2>  
    <h:outputText value="#{timeBean.time}"  
      id="timeDisplay"/>  
  </h2>  
</h:form>
```

The `getTime` method returns different results at different times.

32

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

a4j:poll: Bean

```
public class TimeBean {
    public Date getTime() {
        return(new Date());
    }
}
```

33

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

a4j:poll: faces-config.xml

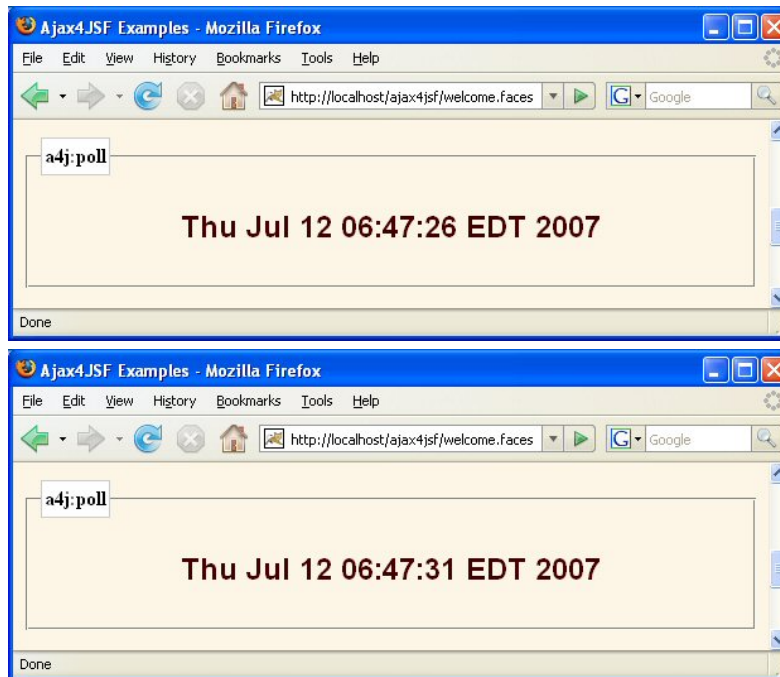
```
<managed-bean>
  <managed-bean-name>
    timeBean
  </managed-bean-name>
  <managed-bean-class>
    coreservlets.TimeBean
  </managed-bean-class>
  <managed-bean-scope>
    application ←
  </managed-bean-scope>
</managed-bean>
```

TimeBean has no state, so for efficiency, reuse the same instance.

34

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

a4j:poll: Results



35

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

a4j:support: Basic Syntax

```
<h:someTag ...>  
  <a4j:support event="javascript-event"  
              reRender="some-id">  
</h:someTag>
```

- **Interpretation**

- Do whatever someTag normally does, but if the specified JavaScript event (onclick, onchange, onkeypress, etc.) occurs, send a behind-the-scenes asynchronous HTTP request to the server that results in the specified JSF element being re-evaluated and re-displayed
- Note: form values are sent and setter methods are run

36

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

JavaScript Event Handlers used with a4j:support

- **onchange**
 - User changes element (IE: and element loses focus)
- **onclick/onclick**
 - User single/double clicks form element or link
- **onfocus/onblur**
 - Element receives/loses focus
- **onkeydown/onkeypress/onkeyup**
 - User presses/presses-or-holds/releases a key
- **onmousedown/onmouseup**
 - User presses/releases mouse
- **onmousemove**
 - User moves mouse
- **onmouseover/onmouseout**
 - User moves mouse onto/off area or link
- **onselect**
 - User selects text within a textfield or textarea
- **onsubmit**
 - User submits form

37

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

a4j:support: Example 1

- **Idea**
 - Use h:inputText to make a textfield
 - As the user types into the textfield, copy the value into regular text
- **Approach**
 - Textfield
 - Stores result in myBean.message
 - `<a4j:support event="onkeyup" reRender="output-region"/>`
 - Separate output field
 - `<h:outputText value="#{myBean.message}" id="output-region"/>`
- **Note**
 - Bundled with ajax4jsf-blank
 - <http://www.coreservlets.com/JSF-Tutorial/>

38

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

a4j:support: Example

```
<h:form>
  <table border="1">
    <tr><th>Textfield</th>
      <th>Ajax Value</th>
    </tr>
    <tr><td width="100">
      <h:inputText value="#{myBean.message}">
        <a4j:support event="onkeyup"
          reRender="output-region"/>
      </h:inputText></td>
      <td width="100">
        <h:outputText value="#{myBean.message}"
          id="output-region"/></td>
      </tr>
    </table>
  </h:form>
```

39

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

a4j:support: Bean

```
public class MessageBean {
  private String message;

  public String getMessage() {
    return(message);
  }

  public void setMessage(String message) {
    this.message = message;
  }
}
```

40

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

a4j:support: faces-config.xml

```
<managed-bean>
  <managed-bean-name>
    myBean
  </managed-bean-name>
  <managed-bean-class>
    coreservlets.MessageBean
  </managed-bean-class>
  <managed-bean-scope>
    request
  </managed-bean-scope>
</managed-bean>
```

41

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

a4j:support: Results

Welcome - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/ajax4jsf-blank/welcome.faces

Google

Welcome

This is the *welcome.faces* page for the *jsf-blank* application, with Ajax4JSF support added. Replace this with your own welcome page.

Sample: Repeater

Textfield	Ajax Value
hello	hello

Done

42

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

a4j:support: Example 2

- **Idea**

- Use h:selectOneMenu to make a list of US states
- When the user selects a state, a list of corresponding cities is shown (again, using h:selectOneMenu)
- When city selected, population of that city is displayed

- **Approach**

- State list
 - `<a4j:support event="onchange" reRender="city-list"/>`
- City List
 - `<a4j:support event="onchange" reRender="pop-field"/>`
- Bean
 - **Make it session scoped** so values persist across across multiple submissions (since there are at least two)

43

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

a4j:support: Example

```
<h:form>
  State:
  <h:selectOneMenu value="#{locationBean.state}">
    <f:selectItems value="#{locationBean.states}"/>
    <a4j:support event="onchange" reRender="cityList"/>
  </h:selectOneMenu><br/>
  City:
  <h:selectOneMenu value="#{locationBean.city}"
    disabled="#{locationBean.cityListDisabled}"
    id="cityList">
    <f:selectItems value="#{locationBean.cities}"/>
    <a4j:support event="onchange" reRender="population"/>
  </h:selectOneMenu><br/>
  Population:
  <h:outputText value="#{locationBean.city}"
    escape="false"
    id="population"/></h:form>
```

44

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

a4j:support: Bean

```
public class LocationBean implements Serializable {
    private String state;
    private String city;
    private boolean isCityListDisabled = true;

    public String getState() { return (state); }

    public void setState(String state) {
        this.state = state;
        isCityListDisabled = false;
    }

    public String getCity() { return(city); }

    public void setCity(String city) { this.city = city; }

    public boolean isCityListDisabled() {
        return(isCityListDisabled);
    }
}
```

Make city list disabled initially.

45

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

a4j:support: Bean (Continued)

```
...
public List<SelectedItem> getStates() {
    List<SelectedItem> states =
        new ArrayList<SelectedItem>();
    states.add(
        new SelectedItem("--- Select State ---"));
    for(StateInfo stateData:
        StateInfo.getNearbyStates()) {
        states.add(
            new SelectedItem(stateData.getStateName()));
    }
    return(states);
}
```

Put dummy value at the top of the list so that any real user selection is considered a change

46

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

a4j:support: Bean (Continued)

```
public SelectItem[] getCities() {
    SelectItem[] cities =
        { new SelectItem("--- Choose City ---")};
    if(!isCityListDisabled && (state != null)) {
        for(StateInfo stateData:
            StateInfo.getNearbyStates()) {
            if(state.equals(stateData.getStateName())) {
                cities = stateData.getCities();
                break;
            }
        }
    }
    return(cities);
}
```

Result of form submission
(i.e., value from the state list)

Value is a SelectItem array

47

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

a4j:support: Supporting Class (StateInfo)

```
public class StateInfo {
    private String stateName;
    private SelectItem[] cities;

    public StateInfo(String stateName,
        SelectItem...cities) {
        this.stateName = stateName;
        this.cities = cities;
    }

    public String getStateName() {
        return(stateName);
    }

    public SelectItem[] getCities() {
        return(cities);
    }
}
```

Value is a SelectItem array

48

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

a4j:support: Supporting Class (StateInfo) Continued

```
private static StateInfo[] nearbyStates =
    { new StateInfo("Maryland",
        new SelectItem("<i>unknown</i>",
            "---- Choose City ----"),
        new SelectItem("635815", "Baltimore"),
        new SelectItem("57907", "Frederick"),
        new SelectItem("57698", "Gaithersburg"),
        new SelectItem("57402", "Rockville")),
    ...
```

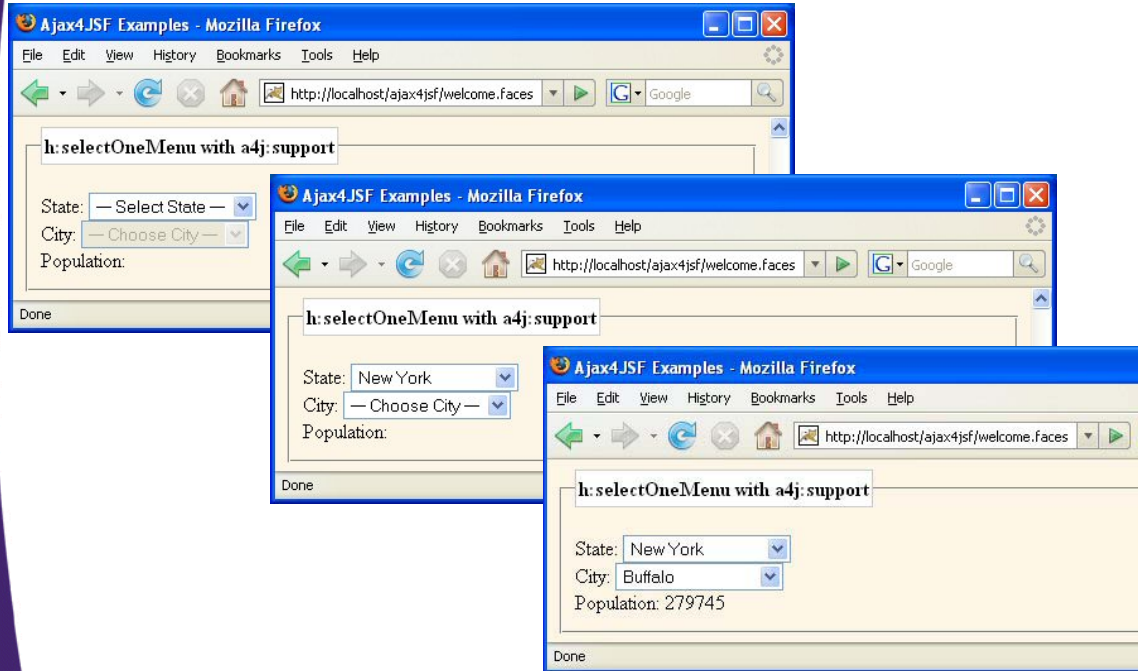
Values that are used when an entry is selected and form is submitted. These are the values displayed in the text when the city field changes

Values that are displayed in the list

a4j:support: faces-config.xml

```
<managed-bean>
  <managed-bean-name>
    locationBean
  </managed-bean-name>
  <managed-bean-class>
    coreservlets.LocationBean
  </managed-bean-class>
  <managed-bean-scope>
    session
  </managed-bean-scope>
</managed-bean>
```

a4j:support: Results



51

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

© 2007 Marty Hall



Other Ajax Tools

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

Servlets, JSP, Struts, JSF/MyFaces, Hibernate, Ajax, GWT, 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.

Tools and Toolkits

- **Client-Side Tools**

- Dojo.
 - Open source JavaScript toolkit with Ajax support
 - <http://www.dojotoolkit.org/>
- Google Web Toolkit
 - Write code in Java, translate it to JavaScript
 - <http://code.google.com/webtoolkit/>
- script.aculo.us
 - Free JavaScript toolkit with Ajax support
 - <http://script.aculo.us/>
- Yahoo User Interface Library (YUI)
 - Free JavaScript toolkit with some Ajax support
 - <http://developer.yahoo.com/yui/>

53

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

Tools and Toolkits (Continued)

- **Server-Side Tools**

- Direct Web Remoting
 - Lets you call Java methods semi-directly from JavaScript
 - <http://getahead.ltd.uk/dwr/>
- JSON/JSON-RPC
 - For sending data to/from JavaScript with less parsing
 - <http://www.json.org/>
 - <http://json-rpc.org/>
- JSP custom tag libraries
 - Create tags that generate into HTML and JavaScript
 - <http://courses.coreservlets.com/Course-Materials/msajsp.html>

54

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

Tools and Toolkits (Continued)

- **Hybrid Client/Server Tools**

- AjaxTags (built on top of script.aculo.us)
 - JSP custom tags that generate Ajax functionality
 - Supports many powerful Ajax capabilities with very simple syntax
 - <http://ajaxtags.sourceforge.net>
- JavaServer Faces (JSF) component libraries
 - Trinidad (formerly Oracle ADF)
 - <http://www.oracle.com/technology/products/jdev/htdocs/partners/addins/exchange/jsf/> (also myfaces.apache.org)
 - Tomahawk
 - <http://myfaces.apache.org/tomahawk/>
 - Ajax4jsf
 - <http://labs.jboss.com/jbossajax4jsf/>
 - IceFaces
 - <http://www.icefaces.org/>
 - Build your own
 - <http://www.coreservlets.com/JSF-Tutorial/>

55

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

Books

- **Foundations of Ajax**
 - Asleson and Schutta. APress.
 - Geared around Java on the server-side.
- **Ajax in Action**
 - Crane, Pascarello, and James. Manning.
 - Geared around Java on the server-side.
- **Pro JSF and Ajax**
 - Jacobi and Fallows. APress.
 - Geared around JavaServer Faces integration.
- **Professional Ajax**
 - Zakas, et al. Wrox. *Wait for 2nd edition.*
 - Geared around Java on the server-side.

56

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

Summary

- **a4j:commandButton/a4j:commandLink**
 - `<a4j:commandButton action="#{bean.method}" value="Button label" reRender="some-id"/>`
 - `<a4j:commandLink action="#{bean.method}" value="Link text" reRender="some-id"/>`
- **a4j:poll**
 - `<a4j:poll interval="x-milliseconds" reRender="some-id"/>`
- **a4j:support**
 - `<h:someTag ...>`
 - `<a4j:support event="javascript-event" reRender="some-id">`
 - `</h:someTag>`

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, GWT, 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.