Lecture 5 – JSP Programming

What Is a JSP Page?

A *JSP page* is a text-based document that contains two types of text: static template data, which can be expressed in any text-based format, such as HTML, SVG, WML, and XML; and JSP elements, which construct dynamic content. A syntax card and reference for the JSP elements are available at

http://java.sun.com/products/jsp/technical.html#syntax

A simple JSP

```
<%--   eg1.jsp          --%>
<%@ page
        language="java"
        import="java.util.*, java.io.*"
%>

<html>
<body>
<%! String message=new String("Hello World"); %>
<p><b><%=message+" How are you doing today?" %></b></p>
</body>
</html>
```

Actually JSP is a Servlet !!!

Application server (says Tomcat) compile a JSP to a Servlet in runtime.

eg1.jsp is compiled to the following Servlet:

```
package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import org.apache.jasper.runtime.*;
import java.util.*;
import java.io.*;

public class eg1_jsp extends HttpJspBase {

 String message=new String("Hello World");

  private static java.util.Vector _jspx_includes;

  public java.util.List getIncludes() {
    return _jspx_includes;
```

HttpJspBase is extended from HttpServlet

```
  }

  public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {

    JspFactory _jspxFactory = null;
    javax.servlet.jsp.PageContext pageContext = null;
    HttpSession session = null;
    ServletContext application = null;
    ServletConfig config = null;
    JspWriter out = null;
    Object page = this;
    JspWriter _jspx_out = null;


    try {
      _jspxFactory = JspFactory.getDefaultFactory();
      response.setContentType("text/html;charset=ISO-8859-1");
      pageContext = _jspxFactory.getPageContext(this, request, response,
                      null, true, 8192, true);
      application = pageContext.getServletContext();
      config = pageContext.getServletConfig();
      session = pageContext.getSession();
      out = pageContext.getOut();
      _jspx_out = out;

      out.write("\r\n");
      out.write("\r\n\r\n");
      out.write("<html>\r\n");
      out.write("<body>\r\n");
      out.write("\r\n");
      out.write("<p>");
      out.write("<b>");
      out.print(message+" How are you doing today?" );
      out.write("</b>");
      out.write("</p>\r\n");
      out.write("</body>\r\n");
      out.write("</html>\r\n\r\n");
    } catch (Throwable t) {
      out = _jspx_out;
      if (out != null && out.getBufferSize() != 0)
        out.clearBuffer();
      if (pageContext != null) pageContext.handlePageException(t);
    } finally {
      if (_jspxFactory != null) _jspxFactory.releasePageContext(pageContext);
    }
  }
}
```

Handling Errors

Any number of exceptions can arise when a JSP page is executed. To specify that theWeb container should forward control to an error page if an exception occurs, include the following page directive at the beginning of your JSP page:

```
<%@ page errorPage="file_name" %>
```

The Duke's Bookstore application page initdestroy.jsp contains the directive

```
<%@ page errorPage="errorpage.jsp"%>
```

The beginning of errorpage.jsp indicates that it is serving as an error page with the following page directive:

```
<%@ page isErrorPage="true|false" %>
```

This directive makes the exception object (of type javax.servlet.jsp.JspException) available to the error page, so that you can retrieve, interpret, and possibly display information about the cause of the exception in the error page.

Implicit Objects

Implicit objects are created by theWeb container and contain information related to a particular request, page, or application. Many of the objects are defined by the Java Servlet technology underlying JSP technology and are discussed at length in Chapter 3. Table 4–2 summarizes the implicit objects.

Variable	Class	Description
application	javax.servlet.ServletContext	The context for the JSP page's servlet and anyWeb components contained in the same application. See Accessing the Web Context (page 75).
config	javax.servlet.ServletConfig	Initialization information for the JSP page's servlet.
exception	java.lang.Throwable	Throwable Accessible only from an error page. See Handling Errors (page 90).
out	javax.servlet.jsp.JspWriter	The output stream.
page	java.lang.Object	The instance of the JSP page's servlet processing the current request. Not typically used by JSP page authors.
pageContext	javax.servlet.jsp.PageContext	The context for the JSP page. Provides a single API to manage the various scoped attributes described in Using Scope Objects (page 55). This API is used extensively when implementing tag handlers (see Tag Handlers, page 125).
request	subtype of	

	javax.servlet.ServletRequest	The request triggering the execution of the JSP page. See Getting Information from Requests (page 60).
response	subtype of javax.servlet.ServletResponse	The response to be returned to the client. Not typically used by JSP page authors.
session	javax.servlet.http.HttpSession	The session object for the client. See Maintaining

Hidden Comment

Documents the JSP page but is not inserted into the response.

JSP Syntax

```
<%-- comment --%>
```

XML Syntax

None.

Example

```
<%@ page language="java" %>
<html>
<head><title>A Comment Test</title></head>
<body>
<h2>A Test of Comments</h2>
<%-- This comment will not be inclueded in the response --%>
</body>
</html>
```

Declaration

Declares a variable or method valid in the scripting language used in the JSP page.

JSP Syntax

```
<%! declaration; [ declaration; ]+ ... %>
```

XML Syntax

```
<jsp:declaration>
declaration; [ declaration; ]+ ...
</jsp:declaration
```

Examples

```
<%! int i = 0; %>
<%! int a, b, c; %>
<%! Circle a = new Circle(2.0); %>
```

Expression

Contains an expression valid in the scripting language used in the JSP page.

JSP Syntax

```
<%= expression %>
```

XML Syntax

```
<jsp:expression>
expression
```

```
</jsp:expression>
```

Examples

```
The map file has <font color="blue"><%= map.size() %></font> entries.
Good guess, but nope. Try
<b><jsp:expression>numguess.getHint()</jsp:expression></b>.
```

Scriptlet

Contains a code fragment valid in the page scripting language.

JSP Syntax

```
<% code fragment %>
```

XML Syntax

```
<jsp:scriptlet>
code fragment
</jsp:scriptlet>
```

Examples

```
<%
String name = null;
if (request.getParameter("name") == null) {
%>
<%@ include file="error.html" %>
<%
} else {
foo.setName(request.getParameter("name"));
if (foo.getName().equalsIgnoreCase("integra"))
name = "acura";
if (name.equalsIgnoreCase( "acura" )) {
%>
```

<jsp:useBean>

Locates or instantiates a bean with a specific name and scope.

JSP Syntax

```
<jsp:useBean id="beanInstanceName"
scope="page|request|session|application"
{
class="package.class" [ type="package.class" ]|
beanName="{package.class | <%= expression %>}"
type="package.class" |
type="package.class"
}
{ /> | > other elements </jsp:useBean> }
```

XML Syntax

```
<jsp:useBean id="beanInstanceName"
scope="page|request|session|application"
{
class="package.class" [ type="package.class" ] |
beanName="{package.class | %= expression %}"
type="package.class" |
type="package.class"
}
{ /> | > other elements </jsp:useBean> }
```

Examples

```
<jsp:useBean id="cart" scope="session" class="session.Carts" />
<jsp:setProperty name="cart" property="*" />
<jsp:useBean id="checking" scope="session" class="bank.Checking" >
<jsp:setProperty name="checking" property="balance" value="0.0" />
</jsp:useBean>
```

Description

The `<jsp:useBean>` element locates or instantiates a JavaBeans component.

`<jsp:useBean>` first attempts to locate an instance of the bean. If the bean does not exist, `<jsp:useBean>` instantiates it from a class or serialized template.

Example of locancal.jsp

```
<%--   loancal.jsp         --%>
<%@ page
       language="java"
       import="java.util.*, java.io.*, ca.on.senecac.ejb605.example.docroot3.*"
%>
<jsp:useBean
       id="loanCalculator"
       scope="page"
       class="ca.on.senecac.ejb605.example.docroot3.LoanCalculator">
  <jsp:setProperty name="loanCalculator" property="principal" value="200000" />
  <jsp:setProperty name="loanCalculator" property="annualRate" value="0.05" />
  <jsp:setProperty name="loanCalculator" property="periodsPerYear" value="12"
/>
  <jsp:setProperty name="loanCalculator" property="years" value="30" />
</jsp:useBean>

<%
  double principal=Double.parseDouble((request.getParameter("principal")));
  double interest=Double.parseDouble((request.getParameter("interest")));
  int periods=Integer.parseInt((request.getParameter("periods")));
  int years=Integer.parseInt((request.getParameter("years")));
%>
<jsp:setProperty name="loanCalculator" property="principal" value="<%=principal
%>" />
<jsp:setProperty name="loanCalculator" property="annualRate" value="<%=interest
%>" />
<jsp:setProperty name="loanCalculator" property="periodsPerYear"
value="<%=periods %>" />
<jsp:setProperty name="loanCalculator" property="years" value="<%=years %>" />

<html>
<body>
<p><b>Loan Calculation</b></p>
<table>
<tr><td>Principal</td><td><%=principal %></td></tr>
<tr><td>Annual Interest</td><td><%=interest %></td></tr>
<tr><td>Periods per year</td><td><%=periods %></td></tr>
<tr><td>Number of years</td><td><%=years %></td></tr>
<tr><td>Payment per period</td><td><%=loanCalculator.computePayment()
%></td></tr>
</table>
</body>
</html>
```

JSP in XML Syntax

Example in JSP format

```
<%--   eg2.jsp             --%>
<%@ page
       language="java"
       import="java.util.*, java.io.*"
%>

<html>
<body>
<p><b>eg2.jsp</b></p>
<table border="2">
<%
  for(int i=0; i<5; i++){
       String aValue="value_"+i;
%>
<tr><td><%=i%></td><td><%=aValue%></td></tr>
<%
  };
%>
</table>
</body>
</html>
```

Example JSP in XML syntax

```
<!-- eg2_xml.jsp  -->
<jsp:root
    xmlns:jsp="http://java.sun.com/JSP/Page"
    version="1.2">
<jsp:directive.page
       language="java"
       import="java.util.*, java.io.*" />
<jsp:declaration>
       String message=new String("Hello World");
</jsp:declaration>
<html>
<body>
<p><b>eg2_xml.jsp</b></p>
<jsp:scriptlet>out.print(message);</jsp:scriptlet>
<table>
<jsp:scriptlet>
       for (int i=0; i&lt;5; i++){
         String aValue="value_"+i;

out.print("&lt;tr&gt;&lt;td&gt;"+i+"&lt;/td&gt;&lt;td&gt;"+aValue+"&lt;/td&gt;&
lt;/tr&gt;");
       };
</jsp:scriptlet>
</table>
</body>
</html>
</jsp:root>
```

Escape Character when using JSP in XML syntax

Entity reference	Stands for...
&	&
'	'
"	"
<	<
>	>