Controlling the Structure of Generated Servlets: The JSP page Directive
Agenda

• Understanding the purpose of the page directive
• Designating which classes are imported
• Specifying the MIME type of the page
• Generating Excel spreadsheets
• Participating in sessions
• Setting the size and behavior of the output buffer
• Designating pages to handle JSP errors
• Controlling threading behavior
Purpose of the page Directive

• Give high-level information about the servlet that will result from the JSP page
• Can control
  – Which classes are imported
  – What class the servlet extends
  – What MIME type is generated
  – How multithreading is handled
  – If the servlet participates in sessions
  – The size and behavior of the output buffer
  – What page handles unexpected errors
The import Attribute

• **By default, the servlet imports:**
  - java.lang.*  javax.servlet.*
  - javax.servlet.jsp.*  javax.servlet.http.*

• **Format**
  - `<%@ page import="package.class" %>`
  - `<%@ page import="package.class1,...,package.classN" %>`

• **Purpose**
  - Generate import statements at top of servlet definition

• **Notes**
  - Only page directive that can appear multiple times within one document
  - Although JSP pages can be almost anywhere on server, classes used by JSP pages must be in normal servlet dirs
  - e.g./WEB-INF/classes or /WEB-INF/classes/directoryMatchingPackage
  - *Always* use packages for utilities that will be used by JSP!
The Importance of Using Packages

- What package will the system think that SomeHelperClass and SomeUtilityClass are in?

```java
... public class SomeClass {
    public String someMethod(...) {
        SomeHelperClass test = new SomeHelperClass(...);
        String someString = SomeUtilityClass.someStaticMethod(...);
        ...
    }
}
```
The Importance of Using Packages

- What package will the system think that SomeHelperClass and SomeUtilityClass are in?

<%  
    SomeHelperClass test = new SomeHelperClass(...);  
    String someString =
        SomeUtilityClass.someStaticMethod(...);  
%>

- If they are in imported packages, there is no ambiguity
- If they are not in packages or if the package to which they belong is not explicitly imported, then the system will assume they are in the same package as the autogenerated servlet
  - This package is not known, different servers use different approaches
The import Attribute: Example (Code)

```java
...<H2>The import Attribute</H2>
<%@ page import="java.util.*,coreservlets.*" %>
<%! 
    private String randomID() {
        int num = (int)(Math.random()*10000000.0);
        return("id" + num);
    }
    private final String NO_VALUE = "<I>No Value</I>";
%>

String oldID =
    CookieUtilities.getCookieValue(request, "userID",
        NO_VALUE);

if (oldID.equals(NO_VALUE)) {
    String newID = randomID();
    Cookie cookie = new LongLivedCookie("userID", newID);
    response.addCookie(cookie);
}

This page was accessed on <%= new Date() %> with a userID cookie of <%= oldID %>.
</BODY></HTML>
```
The import Attribute

This page was accessed on Wed Jun 04 14:50:40 EDT 2003 with a userID cookie of No Value.

The import Attribute

This page was accessed on Wed Jun 04 14:51:10 EDT 2003 with a userID cookie of id4192954.
The contentType and pageEncoding Attributes

• **Format**
  
  - `<%@ page contentType="MIME-Type" %>`
  - `<%@ page contentType="MIME-Type; charset = Character-Set" %>`
  - `<%@ page pageEncoding="Character-Set" %>`

• **Purpose**
  
  - Specify the MIME type of the page generated by the servlet that results from the JSP page

• **Notes**
  
  - Attribute value cannot be computed at request time
  - See section on response headers for table of the most common MIME types
  - `<%@ page contentType="application/vnd.ms-excel" %>` is equivalent to `<%response.setContentType("application/vnd.ms-excel"); %>`
<table>
<thead>
<tr>
<th>First</th>
<th>Last</th>
<th>Email Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marty</td>
<td>Hall</td>
<td><a href="mailto:hall@coreservlets.com">hall@coreservlets.com</a></td>
</tr>
<tr>
<td>Larry</td>
<td>Brown</td>
<td><a href="mailto:brown@coreservlets.com">brown@coreservlets.com</a></td>
</tr>
<tr>
<td>Steve</td>
<td>Balmer</td>
<td><a href="mailto:balmer@ibm.com">balmer@ibm.com</a></td>
</tr>
<tr>
<td>Scott</td>
<td>McNealy</td>
<td><a href="mailto:mcnealy@microsoft.com">mcnealy@microsoft.com</a></td>
</tr>
</tbody>
</table>

<%@ page contentType="application/vnd.ms-excel" %>
<%-- There are tabs, not spaces, between cols. --%>
Conditionally Generating Excel Spreadsheets

• You cannot use the contentType attribute for this task, since you cannot make contentType be conditional.
  – The following always results in the Excel MIME type
    • Attribute values can not be computed at runtime
    • Page directives can not be conditionally inserted as template text

```jsp
<% boolean usingExcel = checkUserRequest(request); %>
<% if (usingExcel) { %>
<%@ page contentType="application/vnd.ms-excel" %>
<% } %>
```

• Solution: use a regular JSP scriptlet with response.setContentType
<BODY>
  <CENTER>
  <H2>Comparing Apples and Oranges</H2>
  <%
  String format = request.getParameter("format");
  if ((format != null) && (format.equals("excel"))) {
    response.setContentType("application/vnd.ms-excel");
  }
  
  <TABLE BORDER=1>
    <TR><TH></TH>         <TH>Apples</TH>  <TH>Oranges</TH></TR>
    <TR><TH>First Quarter</TH>  <TD>2307</TD>  <TD>4706</TD></TR>
    <TR><TH>Second Quarter</TH> <TD>2982</TD>  <TD>5104</TD></TR>
    <TR><TH>Third Quarter</TH>   <TD>3011</TD>  <TD>5220</TD></TR>
    <TR><TH>Fourth Quarter</TH>  <TD>3055</TD>  <TD>5287</TD></TR>
  </TABLE>
  
  %>
  </CENTER></BODY></HTML>
Conditionally Generating Excel Spreadsheets (Results)

Comparing Apples and Oranges

<table>
<thead>
<tr>
<th></th>
<th>Apples</th>
<th>Oranges</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Quarter</td>
<td>2307</td>
<td>4706</td>
</tr>
<tr>
<td>Second Quarter</td>
<td>2982</td>
<td>5104</td>
</tr>
<tr>
<td>Third Quarter</td>
<td>3011</td>
<td>5220</td>
</tr>
<tr>
<td>Fourth Quarter</td>
<td>3055</td>
<td>5287</td>
</tr>
</tbody>
</table>
The session Attribute

• **Format**
  - `<%@ page session="true" %>` <!-- Default -->
  - `<%@ page session="false" %>`

• **Purpose**
  - To designate that page not be part of a session

• **Notes**
  - By default, it is part of a session
    - A value of true signifies that the predefined variable `session` (of type HttpSession) should be bound to the existing session if one exists; otherwise a new session should be created and bound to `session`
    - A value of false saves memory on the server side if you have a high-traffic site
      - It does not disable session tracking; it prevents the JSP page from creating new sessions for users who don’t already have them
      - To truly save memory, *all* related pages have to set `session` to `false`
The isELIgnored Attribute

• **Format**
  - `<%@ page isELIgnored="false" %>`
  - `<%@ page isELIgnored="true" %>`

• **Purpose**
  - To control whether the JSP 2.0 Expression Language (EL) is ignored (true) or evaluated normally (false).

• **Notes**
  - If your web.xml specifies servlets 2.3 (corresponding to JSP 1.2) or earlier, the default is true
    • But it is still legal to change the default—you are permitted to use this attribute in a JSP 2.0 compliant server regardless of the web.xml version.
  - If your web.xml specifies servlets 2.4 (corresponding to JSP 2.0) or earlier, the default is false
The buffer Attribute

• **Format**
  - `<%@ page buffer="sizekb" %>`
  - `<%@ page buffer="none" %>`

• **Purpose**
  - To give the size of the buffer used by the `out` variable

• **Notes**
  - Buffering lets you set HTTP headers even after some page content has been generated (as long as buffer has not filled up or been explicitly flushed)
  - Servers are allowed to use a larger size than you ask for, but not a smaller size
  - Default is system-specific, but must be at least 8KB
The errorPage Attribute

• **Format**
  - `<%@ page errorPage="Relative URL" %>`

• **Purpose**
  - Specifies a JSP page that should process any exceptions thrown but not caught in the current page

• **Notes**
  - The exception thrown will be automatically available to the designated error page by means of the "exception" variable
  - The web.xml file lets you specify *application-wide* error pages that apply whenever certain exceptions or certain HTTP status codes result.
    - The errorPage attribute is for *page-specific* error pages
The isErrorPage Attribute

• Format
  - `<%@ page isErrorPage="true" %>`
  - `<%@ page isErrorPage="false" %> <!-- Default -->`

• Purpose
  - Indicates whether or not the current page can act as the error page for another JSP page

• Notes
  - A new predefined variable called exception is created and accessible from error pages
  - Use this for emergency backup only; explicitly handle as many exceptions as possible
    • Don't forget to always check query data for missing or malformed values
Error Pages: Example

ComputeSpeed.jsp
...
<BODY>
<%@ page errorPage="/WEB-INF/SpeedErrors.jsp" %>
<TABLE BORDER=5 ALIGN="CENTER">
  <TR><TH CLASS="TITLE">Computing Speed</TABLE>
<%!
private double toDouble(String value) {
  return(Double.parseDouble(value));
}
%
<%
double furlongs = toDouble(request.getParameter("furlongs"));
double fortnights =
  toDouble(request.getParameter("fortnights"));
double speed = furlongs/fortnights;
%
<UL>
  <LI>Distance: <%= furlongs %> furlongs.
  <LI>Time: <%= fortnights %> fortnights.
  <LI>Speed: <%= speed %> furlongs per fortnight.
</UL>
</BODY></HTML>
SpeedErrors.jsp

...<BODY>
<%@ page isErrorPage="true" %>
<TABLE BORDER=5 ALIGN="CENTER">
    <TR><TH CLASS="TITLE">
        Error Computing Speed</TABLE>
<P>
ComputeSpeed.jsp reported the following error: <I><%= exception %></I>. This problem occurred in the following place:
<pre>
<%@ page import="java.io.*" %>
<% exception.printStackTrace(new PrintWriter(out)); %>
</pre>
</BODY></HTML>
Error Pages: Results

Computing Speed

- Distance: 12.0 furlongs.
- Time: 3.2 fortights.
- Speed: 3.75 furlongs per fortight.

Error Computing Speed

ComputeSpeed.jsp reported the following error: java.lang.NumberFormatException: For input string: "three". This problem occurred in the following place:

```
java.lang.NumberFormatException: For input string: "three"
at java.lang.NumberFormatException.forInputString(NumberFormatException.java:48)
at java.lang.Float.valueOf(NumberFormatException.java:1212)
at java.lang.Float.parseFloat(Double.parseDouble.java:202)
at org.apache.jsp.ComputeSpeed.jsp.toDouble(ComputeSpeed.jsp.java:15)
at org.apache.jsp.ComputeSpeed.jsp_jspService(ComputeSpeed.jsp.java:66)
```
The isThreadSafe Attribute

- **Format**
  - `<%@ page isThreadSafe="true" %>` <!-- Default --%>
  - `<%@ page isThreadSafe="false" %>`

- **Purpose**
  - To tell the system when your code is not threadsafe, so that the system can prevent concurrent access
    - Normally tells the servlet to implement SingleThreadModel

- **Notes**
  - Avoid this like the plague
    - Causes degraded performance in some situations
    - Causes incorrect results in others
  - **Use explicit synchronization**
Example of Non-Threadsafe Code (IDs Must Be Unique)

• **What's wrong with this code?**
  – A thread could be pre-empted after reading idNum but before updating it yielding two users with the same user ID

```<%! private int idNum = 0; %>
<%
String userID = "userID" + idNum;
out.println("Your ID is " + userID + ".");
idNum = idNum + 1;
%>```
Explicit synchronization

```java
%! private int idNum = 0; %>
<%
synchronized(this) {
    String userID = "userID" + idNum;
    out.println("Your ID is " + userID + ".");
    idNum = idNum + 1;
}
%>

- Better performance in high-traffic environments
  - `isThreadSafe="false"` will totally fail if server uses pool-of-instances approach
The extends Attribute

• **Format**
  – `<%@ page extends="package.class" %>`

• **Purpose**
  – To specify superclass of the servlet that will result from JSP page

• **Notes**
  – Use with extreme caution
  – Can prevent system from using high-performance custom superclasses
  – Typical purpose is to let you extend classes *that come from the server vendor* (e.g., to support personalization features), not to extend your own classes.
Summary

• **The import attribute**
  – Changes the packages imported by the servlet that results from the JSP page
  • *Always* use packages for utility classes!

• **The contentType attribute**
  – Specifies MIME type of result
  – Cannot be used conditionally
  • Use `<% response.setContentType(...); %>` for conditional

• **The errorPage and isErrorPage attributes**
  – Specifies "emergency" error handling pages

• **The isThreadSafe attribute**
  – Turns off concurrent access
  – Always use explicit synchronization instead
Putting it together

CreateCookie.JSP

```html
<HTML>
<HEAD><TITLE>Creating cookies</TITLE></HEAD>
<BODY>
Use the following form to set cookies.<br><br>
<form action=CreateCookie.jsp method=post>
  Cookie Name: <input type=text name=cookieName><br>
  Cookie Value: <input type=text name=cookieValue><br>
  Set maxAge (in seconds): <input type=text name=cookieAge size=10><br>
  <input type=submit value="Submit">
</form>
<%
  String cookieName = request.getParameter("cookieName");
  String cookieValue = request.getParameter("cookieValue");
  if (cookieName !=null && cookieName.length() !=0) {
    Cookie cookie = new Cookie(cookieName,cookieValue);
    int maxAge = 24*60*60;
    try{
      maxAge = Integer.parseInt(request.getParameter("cookieAge"));
      cookie.setMaxAge(maxAge);
    }
    catch(Exception e) {}  
    response.addCookie(cookie);
  }
%
<a href="ListCookie.jsp">List cookies</a><br>
<a href="RemoveCookie.jsp">Delete cookies</a>
</BODY>
</HTML>
```
<HTML>
<HEAD><TITLE>Listing cookies</TITLE></HEAD>
<BODY>
<table border=1>
<tr><th colspan=2 align=center>Cookie List</th></tr>
<tr><th align=center>Cookie Name</th><th align=center>Cookie Value</th></tr>
<% Cookie[] cookies = request.getCookies();
for (int i=0; i<cookies.length; i++){
%>
<tr><td align=center><%=cookies[i].getName()%></td>
<td align=center><%=cookies[i].getValue()%></td></tr>
<% } %></table><br>
<a href="CreateCookie.jsp">Create cookies</a><br>
<a href="RemoveCookie.jsp">Delete cookies</a></BODY>
</HTML>
<HTML>
<HEAD><TITLE>Deleting cookies</TITLE></HEAD>
<BODY>
<% 
    String cookienametoremove = request.getParameter("removecookie");
    Cookie[] cookies = request.getCookies();
    if (cookienametoremove != null) {
        for (int i=0; i<cookies.length; i++) {
            if (cookies[i].getName().equals(cookienametoremove)) {
                cookies[i].setMaxAge(0);
                response.addCookie(cookies[i]);
                response.sendRedirect("RemoveCookie.jsp");
                break;
            }
        }
    }
%>
### RemoveCookie.JSP

```html
<table border=1>
  <tr><th colspan=3 align=center>Cookie List</th></tr>
  <tr><th align=center>Cookie Name</th>
       <th align=center>Cookie Value</th>
       <th> Remove?</th></tr>
  <tr>
    <% for(int i=0; i<cookies.length; i++) { %>
      <td align=center><%= cookies[i].getName() %></td>
      <td align=center><%= cookies[i].getValue() %></td>
      <td> <a href="RemoveCookie.jsp?removecookie=<%= cookies[i].getName()%>">
                  Remove?</a></td>
    </tr>
  <% } %>
</table>

<a href="CreateCookie.jsp">Create cookies</a>
<br>
<a href="ListCookie.jsp">List cookies</a>
</BODY>
</HTML>
```