Advanced Internet Technology Lab

Lab # 9

JSP CONCEPTS

Eng. Haneen El-Masry

April, 2014
Objectives

➢ To be familiar with JSP concepts.

What is JSP?

➢ Java Server Pages (JSP) is a technology to create dynamic webpages.
➢ It enables you to mix regular, static HTML with dynamically generated content.
➢ The regular HTML is written in the normal manner.
➢ The HTML is known as Template Text.
➢ The code for the dynamic parts is enclosed in special tags, most of which start with <%( and end with %>).
➢ JSP files must have the .jsp extension.
➢ You can think of servlets as Java code with HTML inside; but JSP as HTML with Java code inside.

JSP Life Cycle

1- The JSP page is translated into a servlet.
2- Compiled to class.
3- Follow the normal rules for servlets:
   ➢ Loading into memory.
   ➢ Initialization: calling jspInit method.

Note: All the previous steps executed only the first time the JSP page is accessed after having been modified.

➢ Execution: calling service method that calls _jspService method.
➢ Destruction: calling jspDestroy method.

Basic Syntax of JSP

➢ HTML Text

HTML content to be passed unchanged to the client.

<H1>JSP Concepts</H1>
Advanced Internet Technology Lab

- **HTML Comments**

  HTML comment that is sent to the client but not displayed by the browser.

  ```html
  <!-- JSP -->
  ```

- **Template Text**

  Text sent unchanged to the client. HTML text and HTML comments are just special cases of this.

- **JSP Comment**

  Developer comment that is not sent to the client.

  ```jsp
  <!-- JSP -->
  ```

- **Escaped Template Text**

  Text that would otherwise be interpreted specially. Slash is removed and the remaining text is sent to the client.

  ```html
  <\%
  %>
  ```

- **JSP Scripting Elements**

  JSP scripting elements let you insert Java code into the servlet that will be generated from the JSP page. There are three forms:

  1- **JSP Expression**

  Expression that is evaluated and sent to the client each time the page is requested, and inserted into the servlet’s output.

  ```jsp
  <%= Java Value %>
  ```

  2- **JSP Scriptlet**

  Statement or statements that are executed each time the page is requested, which are inserted into the servlet’s _jspService method.

  ```jsp
  <% Java Statement %>
  ```
3- JSP Declaration

Field or method that becomes part of class definition when page is translated into a servlet.

```jsp
<%! Field/ Method Definition %>
```

**JSP Predefined Variables**

To simplify using JSP Expressions and scriptlets, you can use a number of predefined variables as:

- **request**: the HttpServletRequest.
- **response**: the HttpServletResponse.
- **session**: the HttpSession associated with the request (unless disabled with the session attribute of the page directive).
- **out**: the Writer (a buffered version of type JspWriter) used to send output to the client.
- **application**: the ServletContext. This is a data structure shared by all servlets and JSP pages in the Web application and is good for storing shared data.

**Examples**

Random class:

```java
package Package1;

public class Random {

    public static int randint(int range) {
        return (1 + ((int)(Math.random() * range)));
    }
}
```
JSP page using JSP Expression.

```html
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<H1>Random Numbers</H1>
<UL>
  <LI>%=Package1.Random.randomInt(10) %
  <LI>%=Package1.Random.randomInt(10) %
  <LI>%=Package1.Random.randomInt(10) %
  <LI>%=Package1.Random.randomInt(10) %
  <LI>%=Package1.Random.randomInt(10) %
</UL>

</body>
</html>
```

The Result

![Random Numbers]

Using JSP Scriptlet

```html
<H1>Random Numbers</H1>
<UL>
  <% for(int i=0; i<5; i++){
    out.print("<li">"+Package1.Random.randomInt(10)+"</li"/>
  %}>
</UL>
</body>
</html>
```
The Result

Using Declaration

<body>
<%! int i = Package1.Random.randomInt(10);%>

<H1>Random Number is </H1>
<%=i%>
</body>
</html>

The Result

JSP Expression Language Element

Shorthand JSP expression.

\$\{ EL Expression \}

JSP Directive

A JSP directive affects the overall structure of the servlet that results from the JSP page.

<%@ directive attribute1="value1" attribute2="value2" ... attributeN="valueN" %>
In JSP, there are three main types of directives: page, include, and taglib.

1- **Page Directive**

Gives high-level information about the servlet that will result from the JSP page. It can appear anywhere within the document, also it 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.

**Page Directive Attributes**

- **import attribute**

  It lets you specify the packages that should be imported by the servlet into which the JSP page gets translated.

  `<%@ page import="package.class1,...,package.classN" %>`

**Notes:**

- The import attribute is the only page attribute that is allowed to appear multiple times within the same document.
- It is traditional to place import statements either near the top of the document or just before the first place that the referenced package is used.

- **contentType Attribute**

  It sets the Content-Type response header, indicating the MIME type of the document being sent to the client.

  `<%@ page contentType="MIME-Type" %>`

- **pageEncoding Attribute**

  It changes the character set.

  `<%@ page pageEncoding="Character-Set" %>`
- **session attribute**

The session attribute controls whether the page participates in HTTP sessions.

```jsp
<%@ page session="true" %>
<%@ page session="false" %>
```

- **isELIgnored attribute**

The isELIgnored attribute controls whether the JSP 2.0 Expression Language (EL) is ignored (true) or evaluated normally (false).

```jsp
<%@ page isELIgnored="false" %>
<%@ page isELIgnored="true" %>
```

- **buffer Attribute**

The buffer attribute specifies the size of the buffer used by the out variable.

```jsp
<%@ page buffer="sizekb" %>
<%@ page buffer="none" %>
```

**Notes:**

- Buffering lets you set HTTP headers even after some page content has been generated.
- 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.

- **autoFlush Attribute**

The autoFlush attribute controls whether the output buffer should be automatically flushed when it is full (the default) or whether an exception should be raised when the buffer overflows (autoFlush="false").

```jsp
<%@ page autoFlush="true" %>
<%@ page autoFlush="false" %>
```

- **extends Attribute**

To specify a parent class of servlet that will result from JSP page.

```jsp
<%@ page extends="package.class" %>
```
- **isThreadSafe Attribute**

To tell the system when your code is not threadsafe, so that the system can prevent concurrent access.

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

- **errorPage Attribute**

Specifies a JSP page that should process any exceptions thrown but not caught in the current page.

```html
%@ page errorPage="Relative URL" %>
```

**Note:**
The exception thrown will be automatically available to the designated error page by means of "exception" variable.

- **isErrorPage Attribute**

Indicates whether or not the current page can act as the error page for another JSP page.

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

**Note:**
A new predefined variable called exception is created and accessible from error pages.

**Example**

- **JSP Form**

```html
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Error Page</title>
</head>
<body>
<h1>Speed Counting</h1>
<form action="speed.jsp">
Distance: <input type="text" name="d" />
<br />
Time: <input type="text" name="t" />
<br />
<input type="Submit" value="OK" />
</form>
</body>
</html>
```
- speed.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Speed</title>
</head>
<body>
<table border="2">
<tr>
<th>Distance</th>
<th>Time</th>
<th>Speed</th>
</tr>
<td><%= d %></td>
<td><%= t %></td>
<td><%= speed %></td>
</table>
</body>
</html>
```

- SpeedError.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Error Page</title>
</head>
<body>
<%@ page iserrorPage="true" %>
<%@ page import="java.io.*" %>
<center><h1>Error Page</h1></center>
speed.jsp reported the following error: <br>
<% exception.printStackTrace(new PrintWriter(out)); %>
This problem occurred in the following place: <br>
<% exception.printStackTrace(new PrintWriter(out)); %>
</body>
</html>
```
The Output

Speed Counting

Distance: 10
Time: 5
OK

Speed Counting

Distance: 10
Time: 
OK

Error Page

speed.jsp reported the following error:

java.lang.NumberFormatException: empty String

This problem occurred in the following place:
2- Include Directive

It is used to include files at Page translation time, to reuse JSP content in multiple pages, where JSP content affects main page.

```jsp
<%@ include file="Relative URL" %>
```

- JSP Action

Action that takes place when the page is requested.

```jsp
<jsp:action>...</jsp:action>
```

1- Include Action

It lets you insert any of the following into the JSP output:

- The content of an HTML page.
- The content of a plain text document.
- The output of JSP page.
- The output of a servlet.

```jsp
<jsp:include page="Relative URL"/>
```

Note:
JSP content cannot affect main page.

Example

- **whatsnew.jsp**

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1">
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>What's New at JspNews.com</h1>
Here is a summary of our three most recent news stories:
<ol>
<li><jsp:include page="Item1.html" /></li>
<li><jsp:include page="Item2.html" /></li>
</ol>
</body>
</html>
```
- **Item1.html**

```html
<B>Bill Gates acts humble.</B>
In a startling and unexpected development, Microsoft big wig Bill Gates put on an open act of humility yesterday. <br>
<A HREF="http://www.microsoft.com/Never.html">More details...</A>
```

- **Item2.html**

```html
<B>Scott McNealy acts serious.</B>
In an unexpected twist, wisecracking Sun head Scott McNealy was sober and subdued at yesterday's meeting. <br>
<A HREF="http://www.sun.com/Imposter.html">More details...</A>
```

- **The Output**

```
What's New at JspNews.com

Here is a summary of our three most recent news stories:

1. **Bill Gates acts humble.** In a startling and unexpected development, Microsoft big wig Bill Gates put on an open act of humility yesterday.  
   <A HREF="http://www.microsoft.com/More.html">More details...</A>
2. **Scott McNealy acts serious.** In an unexpected twist, wisecracking Sun head Scott McNealy was sober and subdued at yesterday's meeting. 
   <A HREF="http://www.sun.com/More.html">More details...</A>
```

**The jsp:param Element**

The included page uses the same request object as the originally requested page. As a result, the included page normally sees the same request parameters as the main page. If, however, you want to add to or replace those parameters, you can use the `jsp:param` element.

```html
<jsp:param name="" value="" />
```
Example

- **mainPage.jsp**

```html
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Include Action</title>
</head>
<body>
<h1>Parameters of the main page</h1>

FgColor: <%= request.getParameter("fgcolor") %> <br>
BgColor: <%= request.getParameter("bgcolor") %>

<jsp:include page="includedPage.jsp">
<jsp:param name="bgcolor" value="red"/>
</jsp:include>
</body>
</html>
```

- **includedPage.jsp**

```html
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Parameters of included page</h1>

FgColor: <%= request.getParameter("fgcolor") %> <br>
BgColor: <%= request.getParameter("bgcolor") %>

</body>
</html>
```
The Output

2- jsp:forward Action

To obtain the complete output from the auxiliary page.

\[
\text{\textless jsp:forward page=\"<%= \text{destination} %>\textgreater} /\
\]

Difference between redirect and jsp:forward

- redirect sets the response status to 302, and the new URL in a Location header, and sends the response to the browser. Then the browser, according to the http specification, makes another request to the new URL.

- forward happens entirely on the server. The servlet container just forwards the same request to the target URL, without the browser knowing about that. Hence you can use the same request attributes and the same request parameters when handling the new URL. And the browser won't know the URL has changed (because it has happened entirely on the server).