Java server Pages (JSP)

What Is a JSP Page?

- JSP = Java Server Pages
- Combines Java and HTML to create dynamic (changing) web pages
- Similar technologies: ASP, PHP, Perl, Cold Fusion, etc.

- But JSP is the only one in Java!

- A JSP page is a text document that contains two types of text: static data, which can be expressed in any text-based format (such as HTML, SVG, WML, and XML), and JSP elements, which construct dynamic content.
- The recommended file extension for the source file of a JSP page is .jsp. The page can be composed of a top file that includes other files that contain either a complete JSP page or a fragment of a JSP page. The recommended extension for the source file of a fragment of a JSP page is .jspf.

JSP Technology

- JSP is a technology created by Sun Microsystems
- JSPs are closely tied with servlets
- JSP was first announced in the spring of 1998
- JSP is a specification, not a product, and requires support from the server in order to work
- Apache Tomcat server has engine named Jasper
- JSP is a core component of java 2, Enterprise Edition(J2EE)

Purpose of JSPs

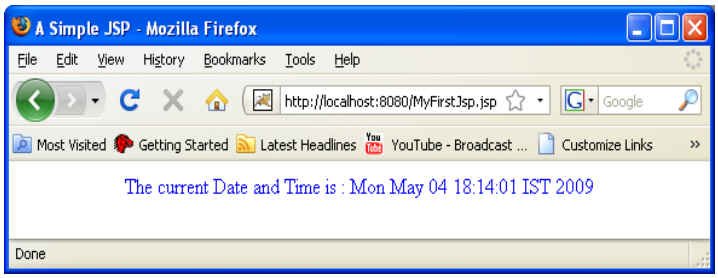
- According to the JSP Specification JSP was expected to;
  - ✓ Enable the separation of dynamic and static content
  - ✓ The authoring of Web pages that create dynamic content easily but with maximum power and flexibility

Using Java Server Pages

- At its most basic, JSP allows for the direct insertion of servlet code into an otherwise static HTML file
- Each block of servlet code (called a scriptlet) is surrounded by a leading <%tag and a closing %> tag
- Any java syntax is valid within these scriptlet tags
- Core JSP classes resides under the javax.servlet.jsp package

A simple JSP

```
<HTML>
<HEAD><TITLE>A Simple JSP</TITLE></HEAD>
<BODY>
  <FONT COLOR = "Blue" FACE = "Trebuchet">
  <CENTER>
 The current Date and Time is : <%= new java.util.Date() %>
  </CENTER>
  </FONT>
</BODY>
</HTML>
```



Example JSP with scriplet

```
<html>
<body>
<%
//any java syntax
%>
 <body>
<html>
```

JSP Elements

The JSP specification defines HTML-like or XML tags that enclose the code in JSP. Thos tags come in three categories :

- ✓ Directive elements - page , include and taglib.
- ✓ Scripting elements - Declarations , Scriptlets and Expressions
- ✓ Action elements

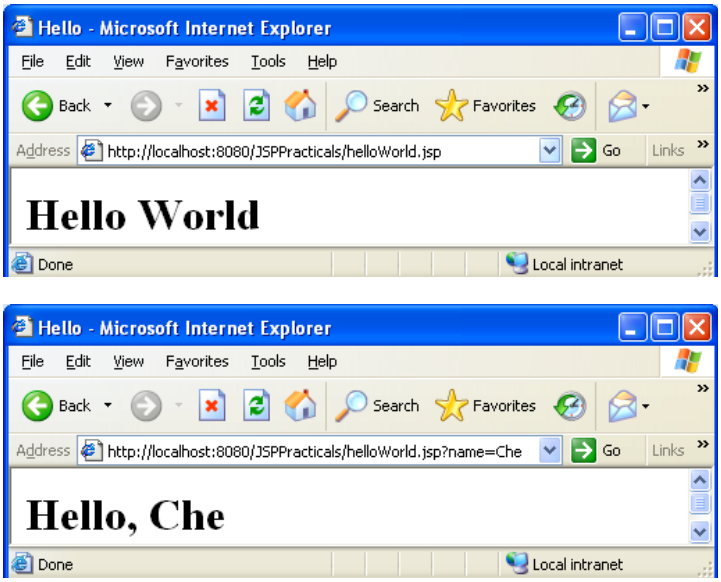
Syntax Elements	Standard Syntax	XML Syntax
Comments	<%--.. --%>	<!-- .. -->
Declarations	<%! ..%>	<jsp:declaration> .. </jsp:declaration>
Directives	<%@ include .. %>	<jsp:directive.include .. />
	<%@ page .. %>	<jsp:directive.page .. />
	<%@ taglib .. %>	xmlns:prefix="tag library URL"
Expressions	<%= ..%>	<jsp:expression> .. </jsp:expression>
Scriptlets	<% ..%>	<jsp:scriptlet> .. </jsp:scriptlet>

Predefined Goodies

- Servlet container makes available the following 6 predefined variables in JSP for our convenience.
  - ✓ HttpServletRequest request-the servlet request
  - ✓ HttpServletResponse response-the servlet response
  - ✓ Javax.servlet.jsp.JspWriter out-the output writer
  - ✓ HttpSevletContext application- the web application
  - ✓ HttpSession session –the users session
  - ✓ javax.servlet.jsp.PageContext pageContext- an object primarily used to abstract the implementation of server

Exercise: Write a JSP code to get request parameter “name” and display Hello along with the name extract form the request. If no value sent with the request display “Hello World”. Use scriplet only

```
<%-- hello.jsp --%>
<HTML>
<HEAD><TITLE>Hello</TITLE></HEAD>
<BODY>
<H1>
<%
if (request.getParameter(“name”)==null){
out.println(“Hello World”);
}
else{
out.println(“Hello , “+
request. getParameter(“name”) ) ;
}
%>
</H1>
</BODY>
</HTML>
```

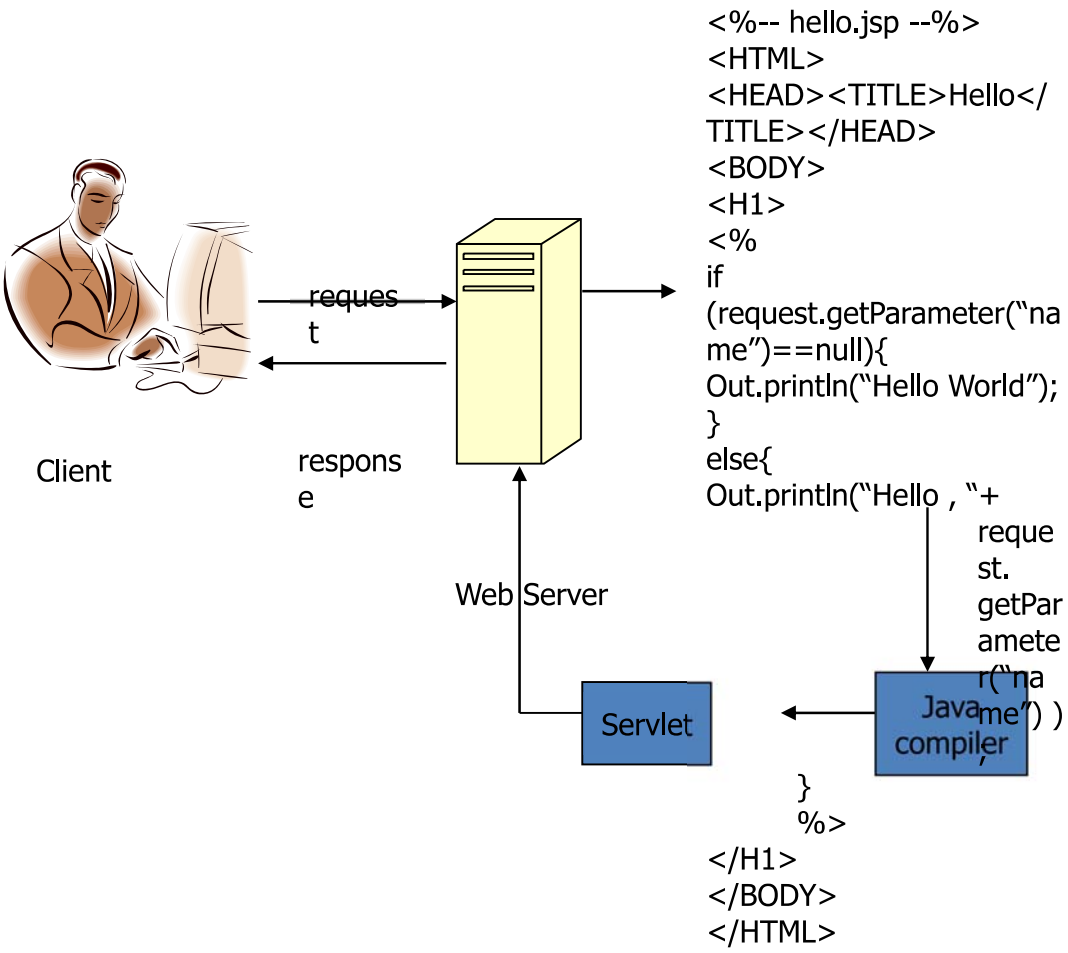


What makes JSP work?

- Behind the screen, the server automatically creates, compiles, loads and runs a special servlet to generate the page’s content.
- The static portion of the HTML page are generated by this special servlet using the equitant of out.prinln() calls ,while the dynamic portion are included directly

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
```

```
public class HelloUser2
   extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = new PrintWriter(response.getOutputStream
    String sName = request.getParameter("name");
……..
```



The "first-person penalty"

- The first time you access a JSP page the server will create and compile the background servlet, hence lagging the response a little bit
- You can avoid the first person penalty by pre-compiling your JSPs
- This can be done manually by making a request to each JSP page passing a query string of ?jsp-precompile=true –this request server to compile the JSPs but the special query string tells the server not to bother passing the request through to the JSP for handling-

I. JSP comments

You can put two different types of comments

<!-- HTML comment -->

The container just passes this on to the client where the browser interprets it as a comment

<%--JSP Comment--%>

These are for page developers, just like java comments in a java source file.

II. JSP Directives

- A JSP directive allows a JSP page to control certain aspects of the background jsp processing servlet
- Directive is a way for you to give special instructions to the container at page translation time.
- Types of directives
  a. page
  b. include

c. taglib

- Directive syntax is a name-value pair

<%@ directiveName attribName="attribValue" %>

a. Page Directives

- The page directive allows the JSP to control the generated servlet through the setting of special attributes
- The page directive's JSP style form is :

<@ page attributes %>

- The following table shows the common page directive attributes

Attribute	Description
import	Specifies a list of classes and packages the generated servlet should import. Multiple classes can be given in a comma-separated list. Eg: <% @ page import="java.io.*, java.util.HashTable" %>
session	Indicates the page wants to have access to the user's session. A "true" puts the session variable in scope and may set a client cookie to manage the session. A "false" disables access to the session variable. For example ; <% @ page session="false" %> The default is true.
isThreadSafe	
errorPage	Specifies a URL of the error page that will be used to handle exception. This proves useful because it's difficult to do try/catch blocks when writing JSP pages. Eg: <% @ page errorPage="/error.jsp" %> When you do not provide a URL , the container can use its own default.
isErrorPage	Indicates the page is intended to be used as the target of an errorPage. If the value is true the page can access an implicit variable named exception to retrieve the Throwable. Eg: <% @ page isErrorPage="false" %> The default value for this attribute is false.
contentType	Specifies the content type of the generated page. Eg: <% @ page contentType="text/plain" %> The default content type is text/html; charset=8859_1
pageEncoding	The character set of the current page. The default is ISO-8859-1 (Latin script) for JSP-style and UTF-8(an 8-bit Unicode encoding) for XML-style tags.

Eg :

```
<%@ page import="java.util.GregoreanCalender" %>
<%!
GregoreanCalender gc=new GregoreanCalender();
%>
<html>
<body>
<h2><%=gc.getTime()%</h2>
</body>
</html>
```

b. Include Directives

- The include directive is used to include another page within the current page.
- The include directive's JSP style form is :

<@ include attributes %>

- Using directive

```
<%@ page import="foo.*" %>
<html>
<body>
The page count is:
<%out.println(Counter.getCount());
```

```
%>
</body>
</html>
```

```
<html><body>
Test scriptlets…
<% int y=5+x; %>
<% int x=2; %>
</body></html>

<html><body>
Test scriptlets…….
<%! int x= 42; %>
<% int x=22; %>
<%=x %>
</body></html>
```

Scripting Elements

a. Declarations in JSP

- A declaration begins with <%! And ends with %>
- In between the tags you can include any servlet code that should be placed outside the servlets service method
- You may declare static or instance variables or define new methods

Variable declaration

```
<%! Vector  v = new Vector(); %>
<%! Int x=0; int Y =10; %>
```

- Any variable or method you declare within a declaration element becomes an instance variable of the JSP page implementation class, and therefore is global to the entire page.

Method declaration

```
<html>
<body>
<%! int doubleCount() {
      count=count*2;
      return count;
}
%>
<%! int count=1;%>
The page count is now:
<%=doubleCount() %>
</body>
</html>
```

b. Sciptlets

- Scriptlets contain java code statements. The code in the scriptlet appears in the translated JSP, but not in the output to the client.
- Regular old java code can be put in a JSP using scriptlets which just means java code within a <% ..... %>
- Eg:

```
<%
      For (int i=0 ;i<10 ; i++){
%>Hello World !
<%
      }
%>
```

```
<html>
<body>
The page count is:
<%
out.println(foo.Counter.getCount());
%>
</body>
</html>


package foo;
public class Counter{
private static int count;
public static int getCount(){
count++;
return  count;
}
}
```

c. Expressions in JSP

- A JSP expression begins with <%= and ends with %>
- E.g.

  <%="Hello Sam"%>

  <%=salary-deductions%>
- Any java expression between the two tags is evaluated, the result is converted to string and the text is included directly in the page.
- This techniques eliminates the clutter of an out.println() call.(expressions becomes the arguments to an out.println()