2-07-2020

First of all we need to know the software requirements —

① Eclipse IDE (download & install)
— It is open source (free to use)

② Apache Tomcat Server Version 9.0
— It is web server software used to run any web application in our computer.

Steps to create JSP in Eclipse IDE with Tomcat Server :—

① Open Eclipse IDE.

② click on File > New > Dynamic web project —

③ Type the name of your project name and click on Target runtime
click on Dynamic Web module Version to Select Version

④ click on Next button >> Next

⑤ click on ☐ Generate web.xml deployment descriptor

⑥ click on Finish button.

⑦ Create the JSP file in eclipse IDE
ⓐ click on + icon of project
ⓑ Right click on Sub-folder WebContent
ⓒ click New > JSP > Type filename >> Next >> Finish.

⑧ Let us write some code in the body section of the JSP file as:—

<h1> Hello and Welcome to JSP </h1>

⑨ To Run the Project — Right click on Project name > Run as >> Run on Server >> choose tomcat server >> next >> add all >> finish.

Note:- To configure Tomcat Server in Eclipse:—

(First time only)

→ Click on Servers tab at bottom side of IDE.

→ Right click on blank area > New > Servers > choose Tomcat then its version > Next > click on Browse button > Select the apache tomcat root folder previous to bin folder > Next > add all > Finish.

Or you can also right click on jsp page > Run as > Run on Server

Scriptlet tags of JSP :— Java codes can be written inside the jsp page using the scriptlet tags. These tags are of three types —

ⓐ Scriptlet tag
ⓑ expression tag
ⓒ declaration tag

ⓐ Scriptlet tag :- These tags are used to execute only java source in jsp page.

Syntax is :- <% java source code %>

ⓑ Expression tag :- It is mainly used to print the values of variable or method.

Syntax is :- <% = statement %>

```
<html><body>
eg:- <% = "Welcome to JSP"%>
</body>
</html>
```

<% = java.util.Calendar.getInstance().getTime() %>

tag can be placed inside the body to print current time.

Example to print username :-

index.jsp

```
<html><body>
<Form action = "welcome.jsp">
<input type = "text" name = "uname"><br>
<input type = "submit" value = "go">
</Form>
</body>
</html>
```

user input stored to uname

welcome.jsp

```
<html><body>
<% = "Welcome" + request.getParameter
     ("uname") %>
</body>
</html>
```

get stored value from uname

(c) JSP Declaration tag — used to declare fields and methods.

The code written inside the declaration tag is placed outside the service() method of auto generated Servlet. So it does not get memory at each request.

Syntax:—

<%! field or method declaration %>

eg:- <%! int cube(int n) { return n*n*n;} %>

<%= "Cube of 5 is :" + cube(5) %>

JSP implicit Objects :—

There are 9 jsp implicit objects created by the web container.

	Name of Object	9 ts Type
①	out	JspWriter
②	request	HttpServletRequest
③	response	HttpServletResponse
④	config	ServletConfig
⑤	application	ServletContext
⑥	session	HttpSession
⑦	pageContext	PageContext
⑧	page	Object
⑨	exception	Throwable

Details of objects:—

1. **out** — for writing any data to the buffer, it is defined in Servlets as:—

```
PrintWriter out = response.getWriter();
```

But in case of JSP, we no need to write any code, just use it.

example — index.jsp

```
<html>
<body>
<% out.print("Today is:" + java.util.Calendar.
        getInstance().getTime()); %>
</body>
</html>
```

2. **request** — request is object of HttpServletRequest created by the web container for each JSP request. It can be used to get request information such as parameter, header information, remote address, server name, server port, content-type, character encoding etc.

Example:— index.html

```
<form action="welcome.jsp">
<input type="text" name="uname">
<input type="submit" value="go"><br/>
</form>
```

welcome.jsp

```
<% String name = request.getParameter("uname");
   out.print("Welcome " + name);
%>
```

3. **response** — It is an implicit object of type HttpServletResponse. The instance of this class is created by the web container for each jsp request.

Example:- index.html

```
<html><head><title> User Response</title>
</head><body><Form action="welcome.jsp">
<input type="text" name="uname">
<input type="submit" value="go"><br/>
</form></body></html>
```

welcome.jsp

```
<% response.sendRedirect("https://www.google.com");
%>
```

4. **config** — It is an implicit object of ServletConfig. This object can be used to get initialization parameter for a particular JSP page.

Generally, it is used in web.xml file to get initial parameter.

Example:- index.html:

```
<html> <head><title> config obj example
</title></head><body>
```

```
<form action="welcome">
<input type="text" name="uname">
<input type="submit" value="go"><br/>
</form>
</body>
</html>
```

web.xml file :—

```
<web-app>
<servlet>
  <servlet-name> shesshah </servlet-name>
  <jsp-file> /welcome.jsp </jsp-file>

<init-param>
  <param-name> drname </param-name>
  <param-value> sun.jdbc.odbc.JdbcOdbcDriver </param-value>

</init-param>

</servlet>

<servlet-mapping>
<servlet-name> shesshah </servlet-name>
<url-pattern> /welcome </url-pattern>
</servlet-mapping>

</web-app>
```

welcome.jsp

```
<% out.print("Welcome" + request.getParameter("uname"));
String driver = config.getInitParameter("drname");
out.print("Driver name is :" + driver); %>
```