SESSION TRACKING

### Cookies

A webserver can assign a unique session ID as a cookie to each web client and for subsequent requests from the client they can be recognized using the received cookie.

This may not be an effective way as the browser at times does not support a cookie. It is not recommended to use this procedure to maintain the sessions.

## The session Object

Apart from the above mentioned options, JSP makes use of the servlet provided HttpSession Interface. This interface provides a way to identify a user across.

* a one page request or
* visit to a website or
* store information about that user

By default, JSPs have session tracking enabled and a new HttpSession object is instantiated for each new client automatically. Disabling session tracking requires explicitly turning it off by setting the page directive session attribute to false as follows −

<%@ page session = "false" %>

The JSP engine exposes the HttpSession object to the JSP author through the implicit **session** object. Since **session** object is already provided to the JSP programmer, the programmer can immediately begin storing and retrieving data from the object without any initialization or **getSession()**.

### Create a directory structures

* Create a dynamic web project.
* Put your HTML file inside web-appfolder
* Create a new java file inside the folder
* After compiling the java file put the class file inside Web-Inf🡪classes directory.

1. Create a deployement descriptor in Web.xml.
2. Put your servlet name and servlet class name in the appropriate tag inside web.xml.
3. Build and Deploy the Project.

**Note:**

1. set JAVA\_HOME or JRE\_HOME in environment variable (It is required to start server).
2. Change the port number of tomcat (optional). It is required if another server is running on same port (8080).

Go to My Computer properties -> Click on advanced tab then environment variables -> Click on the new tab of user variable -> Write JAVA\_HOME in variable name and paste the path of jdk folder in variable value -> ok -> ok -> ok.

INVOKING SERVLET FROM HTML

Sevlet Code:

 import java.io.\*;

 import java.util.\*;

 import javax.servlet.\*;

public class PostParam extends GenericServlet

 {

public void service(ServletRequest request,ServletResponse response) throws

ServletException,IOException

{

PrintWriter pw = response.getWriter();

 Enumeration e = request.getParameterNames();

 while(e.hasMoreElements())

 {

String pname = (String)e.nextElement();

 pw.print(pname + " = ");

 String pvalue = request.getParameter(pname);

 pw.println(pvalue);

 }

 pw.close();

}

}

HTML CODE:

<HTML>

 <head>

 <TITLE>INVOKING SERVLET FROM HTML</TITLE>

 </head>

 <BODY>

 <CENTER>

 <FORM name = "PostParam" method = "Post" action="http://localhost:8080">

 <TABLE>

 <tr>

 <td><B>Employee </B> </td>

 <td><input type = "textbox" name="ename" size="25"

 value=""></td>

 </tr>

 <tr>

 <td><B>Phone </B> </td>

<td><input type = "textbox" name="phoneno" size="25"

value=""></td>

</tr>

</TABLE>

<INPUT type = "submit" value="Submit">

</FORM>

 </CENTER>

 </body>

 </html>