ITL 3 PRACTICAL - 03

**AIM**

**Write a program to pass data using sessions**

**Session in Java**

**The time interval in which two systems (i.e., the client and the server) communicate with each other can be termed as a session. In simpler terms, a session is a state consisting of several requests and response between the client and the server.**

**It is a known fact that HTTP and Web Servers are both stateless. Hence, the only way to maintain the state of the user is by making use of technologies that implement session tracking.  
Session tracking in servlets can be implemented by a number of methods, cookies being one of them. However, they have multiple disadvantages:**

* **Only textual information can be kept by them.**
* **If cookies are disabled by a user, the web application is unable to make use of them.**
* **Not more than 4kb of data can be contained by a single cookie.**
* **Another way to implement session tracking is by creating sessions with unique session ids for every user in a java servlet.**

**Session is a conversional state between client and server and it can consists of multiple request and response between client and server. Since HTTP and Web Server both are stateless, the only way to maintain a session is when some unique information about the session (session id) is passed between server and client in every request and response.**

**There are several ways through which we can provide unique identifier in request and response.**

1. **User Authentication – This is the very common way where we user can provide authentication credentials from the login page and then we can pass the authentication information between server and client to maintain the session. This is not very effective method because it wont work if the same user is logged in from different browsers.**
2. **HTML Hidden Field – We can create a unique hidden field in the HTML and when user starts navigating, we can set its value unique to the user and keep track of the session. This method can’t be used with links because it needs the form to be submitted every time request is made from client to server with the hidden field. Also it’s not secure because we can get the hidden field value from the HTML source and use it to hack the session.**
3. **URL Rewriting – We can append a session identifier parameter with every request and response to keep track of the session. This is very tedious because we need to keep track of this parameter in every response and make sure it’s not clashing with other parameters.**
4. **Cookies – Cookies are small piece of information that is sent by web server in response header and gets stored in the browser cookies. When client make further request, it adds the cookie to the request header and we can utilize it to keep track of the session. We can maintain a session with cookies but if the client disables the cookies, then it won’t work.**
5. **Session Management API – Session Management API is built on top of above methods for session tracking. Some of the major disadvantages of all the above methods are:**
   * **Most of the time we don’t want to only track the session, we have to store some data into the session that we can use in future requests. This will require a lot of effort if we try to implement this.**
   * **All the above methods are not complete in themselves, all of them won’t work in a particular scenario. So we need a solution that can utilize these methods of session tracking to provide session management in all cases.**

## **Http Session Interface**

**Servlets in java provide an interface known as ‘HttpSessionInterface’.  
They consist of various methods, some of which are:**

* **public HttpSession getSession(boolean create): This method gets the session associated with the request. In case it is not available or not present, a new session is created which is based upon the Boolean argument specified.**
* **public String getId(): The unique session id is returned by this method.**
* **public long getCreationTime(): The time when the session was created is returned by this method. It is measured in milliseconds.**
* **public long getLastAccessedTime(): The time when the session was last accessed is returned by this method. It is measured in milliseconds.**
* **public void invalidate(): A session can be invalidated by using this method.**

**CODE**

**package com.journaldev.servlet.session;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import javax.servlet.RequestDispatcher;**

**import javax.servlet.ServletException;**

**import javax.servlet.annotation.WebServlet;**

**import javax.servlet.http.Cookie;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import javax.servlet.http.HttpSession;**

**/\*\***

**\* Servlet implementation class LoginServlet**

**\*/**

**@WebServlet("/LoginServlet")**

**public class LoginServlet extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**private final String userID = "admin";**

**private final String password = "password";**

**protected void doPost(HttpServletRequest request,**

**HttpServletResponse response) throws ServletException, IOException {**

**// get request parameters for userID and password**

**String user = request.getParameter("user");**

**String pwd = request.getParameter("pwd");**

**if(userID.equals(user) && password.equals(pwd)){**

**HttpSession session = request.getSession();**

**session.setAttribute("user", "Pankaj");**

**//setting session to expiry in 30 mins**

**session.setMaxInactiveInterval(30\*60);**

**Cookie userName = new Cookie("user", user);**

**response.addCookie(userName);**

**//Get the encoded URL string**

**String encodedURL = response.encodeRedirectURL("LoginSuccess.jsp");**

**response.sendRedirect(encodedURL);**

**}else{**

**RequestDispatcher rd = getServletContext().getRequestDispatcher("/login.html");**

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

**out.println("<font color=red>Either user name or password is wrong.</font>");**

**rd.include(request, response);**

**}**

**}**

**}**

**<%@ page language="java" contentType="text/html; charset=US-ASCII"**

**pageEncoding="US-ASCII"%>**

**<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "https://www.w3.org/TR/html4/loose.dtd">**

**<html>**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=US-ASCII">**

**<title>Login Success Page</title>**

**</head>**

**<body>**

**<%**

**//allow access only if session exists**

**String user = null;**

**if(session.getAttribute("user") == null){**

**response.sendRedirect("login.html");**

**}else user = (String) session.getAttribute("user");**

**String userName = null;**

**String sessionID = null;**

**Cookie[] cookies = request.getCookies();**

**if(cookies !=null){**

**for(Cookie cookie : cookies){**

**if(cookie.getName().equals("user")) userName = cookie.getValue();**

**if(cookie.getName().equals("JSESSIONID")) sessionID = cookie.getValue();**

**}**

**}else{**

**sessionID = session.getId();**

**}**

**%>**

**<h3>Hi <%=userName %>, Login successful. Your Session ID=<%=sessionID %></h3>**

**<br>**

**User=<%=user %>**

**<br>**

**<!-- need to encode all the URLs where we want session information to be passed -->**

**<a href="<%=response.encodeURL("CheckoutPage.jsp") %>">Checkout Page</a>**

**<form action="<%=response.encodeURL("LogoutServlet") %>" method="post">**

**<input type="submit" value="Logout" >**

**</form>**

**</body>**

**</html>**

**package com.journaldev.servlet.session;**

**import java.io.IOException;**

**import javax.servlet.ServletException;**

**import javax.servlet.annotation.WebServlet;**

**import javax.servlet.http.Cookie;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import javax.servlet.http.HttpSession;**

**/\*\***

**\* Servlet implementation class LogoutServlet**

**\*/**

**@WebServlet("/LogoutServlet")**

**public class LogoutServlet extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**response.setContentType("text/html");**

**Cookie[] cookies = request.getCookies();**

**if(cookies != null){**

**for(Cookie cookie : cookies){**

**if(cookie.getName().equals("JSESSIONID")){**

**System.out.println("JSESSIONID="+cookie.getValue());**

**}**

**cookie.setMaxAge(0);**

**response.addCookie(cookie);**

**}**

**}**

**//invalidate the session if exists**

**HttpSession session = request.getSession(false);**

**System.out.println("User="+session.getAttribute("user"));**

**if(session != null){**

**session.invalidate();**

**}**

**//no encoding because we have invalidated the session**

**response.sendRedirect("login.html");**

**}**

**}**

**OUTPUT**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, text, application, Word

Description automatically generated**