Session tracking

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletContext;

import javax.servlet.ServletException;

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 FirstServlet

\*/

public class FirstServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public FirstServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)

\*/

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

//HttpSession

ServletContext

PrintWriter pw = response.getWriter();

/\* Create the Cookie \*/

Cookie ck = new Cookie("ck1", "cv1");

/\* Attaching the cookie to response object \*/

response.addCookie(ck);

/\* How to retrieve the cookies from request \*/

Cookie[] cklist = request.getCookies();

if(cklist== null) {

pw.println("Welcome.......");

} else {

pw.println("Welcome Back.....");

for(Cookie ck1 : cklist) {

pw.println(ck1.getName() + " :: " + ck1.getValue());

}

}

}

}

Http Session

public abstract long getCreationTime();

used to retrieve the session created time

public abstract java.lang.String getId();

used to retrieve the Unique Id assigned to the session

public abstract long getLastAccessedTime();

used to retrieve the info. About when this session was last accessed.

public abstract javax.servlet.ServletContext getServletContext();

used to get the servlet context

public abstract void setMaxInactiveInterval(int arg0);

used to set the inactive interval. If that configured time interval is elapsed then the servlet container destroys the session.

public abstract int getMaxInactiveInterval();

used to find the max active interval configured

public abstract javax.servlet.http.HttpSessionContext getSessionContext();

used to get the session context.

public abstract void invalidate();

used to destroy the session

public abstract boolean isNew();

used to check whether the current session is a newly created session or existed session

public abstract java.lang.Object getAttribute(java.lang.String arg0);

public abstract java.lang.Object getValue(java.lang.String arg0);

public abstract java.util.Enumeration getAttributeNames();

public abstract java.lang.String[] getValueNames();

public abstract void setAttribute(java.lang.String arg0, java.lang.Object arg1);

public abstract void putValue(java.lang.String arg0, java.lang.Object arg1);

public abstract void removeAttribute(java.lang.String arg0);

public abstract void removeValue(java.lang.String arg0);

}

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.util.Date;

**import** java.util.Enumeration;

**import** javax.servlet.ServletContext;

**import** javax.servlet.ServletException;

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

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

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

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

/\*\*

\* Servlet implementation class HomeServlet

\*/

**public** **class** HomeServlet **extends** HttpServlet {

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

/\*\*

\* **@see** HttpServlet#HttpServlet()

\*/

**public** HomeServlet() {

**super**();

// **TODO** Auto-generated constructor stub

}

/\*\*

\* **@see** HttpServlet#service(HttpServletRequest request, HttpServletResponse response)

\*/

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

HttpSession session = request.getSession();

PrintWriter pw = response.getWriter();

pw.println("Is Session New?" + session.isNew());

pw.println("Session Id :" + session.getId());

pw.println("Session Created Time:" + **new** Date(session.getCreationTime()));

pw.println("Session Last Accessed Time:" + **new** Date(session.getLastAccessedTime()));

ServletContext ctxt = session.getServletContext();

pw.println("Context path = " + ctxt.getContextPath());

pw.println("Servlet names...");

Enumeration<String> servletNames = ctxt.~~getServletNames~~();

**while**(servletNames.hasMoreElements()) {

pw.println(servletNames.nextElement());

}

pw.println("Server info = " + ctxt.getServerInfo());

}

}

How many ways we can invalidate the session?

1. setMaxInactiveInterval (int)

session.setMaxInactiveInterval(40);

1. invalidate()

session.invalidate();

1. configure the max inactive interval in web.xml

<session-config>

<session-timeout>1</session-timeout>

</session-config>