

LoginServlet----- shows the login page
VerifyServlet ---- verify the login
HomeServlet -----shows the home page

1st run ---in all the 3 servlets use the doGet method
2nd run----replace doGet with service method

Session Management in web application---

From **login up to logout** , we want the server to remember the data!!!
By default any web server does not remember the data after request response cycle .
WHY ? Http is a STATELESS protocol.

Through session management we will make the server remember the data
And also keep track of which data comes from which browser(USER) !!
i.e. USERWISE data tracking/maintenance

Cookies are used for User preference !!! Is a way to attract the USER to your site !!!

To maintain session we use an inbuilt API javax.servlet.http.HttpSession
This internally uses cookies.

Lets say I have 3 USERS ---denoted by 3 browsers

Each user will access a URL and refresh it !!!

Server should remember how many times the particular user accessed that URL
----maintain a USERWISE counter

HttpSession session = request.getSession();

getSession API --- checks the request whether already a session ID is coming in the cookies
Then it knows that this is **EXISTING** session
It will retrieve sessionId from the cookie and access the shelf of that
sessionId and return the EXISTING session object

If the request does not have session ID
Then it knows that the **NEW** session has to be created
It will create a new session ID and return a NEW session object

The WEB SERVER maintains one HttpSession object per Session ID .

login uname password

home //should be shown only when logged in

Welcome User

Enter your school name :

Go to another page -----> on another page show User name at the top

Show a button ----yet another page

Show user on the
top

Your
school
name is :

For every user we will save the username and the schoolname!!!

On each page we add a logout link/button

After logout the username and schoolname are discarded

session.getSession()	creates a session always if no session exists
session.getSession(false)	It returns null if no session exists

Rd . Forward	forward the current request to another servlet and then send the output of another servlet to the browser
Response. sendRedirect	a new request is sent to another servlet and the output is sent to browser

for logout	session.invalidate + we removed all the session attributes
------------	--

Servlet ----- **We embed HTML inside a JAVA class**

Writing complicated and big HTMLs inside a JAVA class is very difficult

To be able to write **htmls easily** + and still get the advantage of **dynamic** html + **session** management
+ **DB** connectivity + Java **Logic** Power
JSP was introduced !!!

JSP = Java Server Pages

JSP = **We embed Java code inside HTML**

JSP is **at server side** Just like servlet

JSP is **translated to a Servlet** by Tomcat JSP Engine (JASPER)

Programmer writes JSP , JSP Engine generates Servlet !!!

All in all the concepts of Servlets are used to generate dynamic html , session , DB , Logic

Print Hello 100 times with serial number using JSP !!!

We use a JSP SCRIPLET to include JAVA logic

<% %>

This code is added in the _jspservice method by Jasper !!!

Output Expression <%= %>

We can write a variable name or a method call inside output expression!!!

The value is printed in the output html !!

