

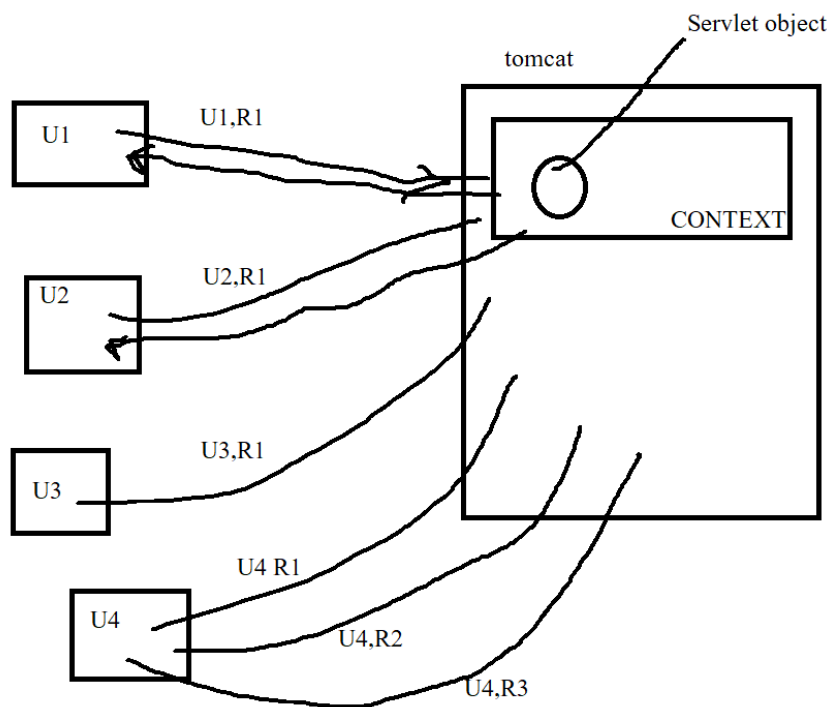
## Session Management -----

Http is a stateless protocol !!!

It does not remember the state(values) of previous request !!

It only knows about current request response

Session Management ---- The Http Server by default does not remember the data of  
PREVIOUS REQUESTS -----FROM A USER



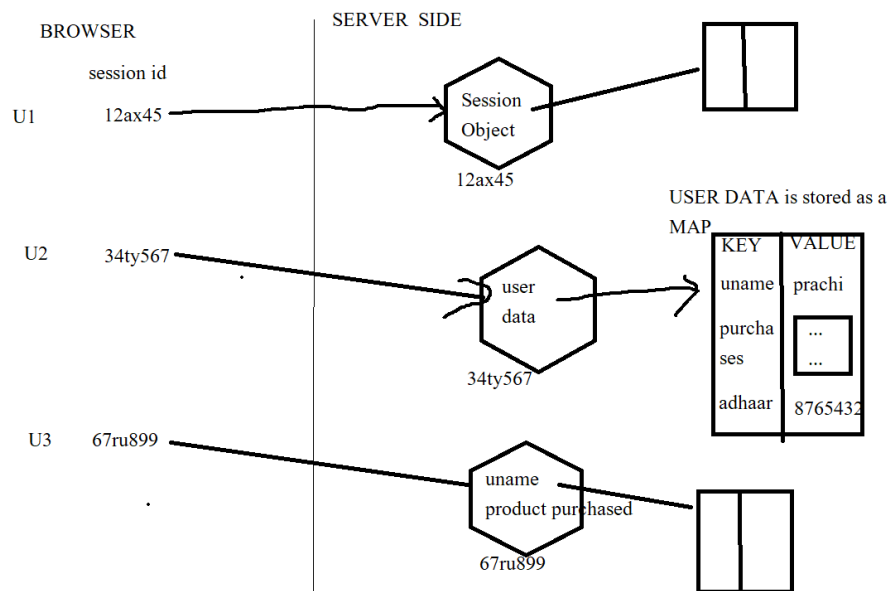
How to Identify a User ?

Browser wise --- by default each browser is treated as a User !!!

Internally the Http Container assigns a UNIQUE ID to each user

This ID is called as **Session ID**

Http Container maintains a Session object for storing data of that USER for each Session ID !!!



WHO creates the Session Id ? Web CONTEXT in tomcat server

How is the session ID sent to the browser from server ?

How is the session ID sent from browser to Server ?

This TOKEN EXCHANGE are done using

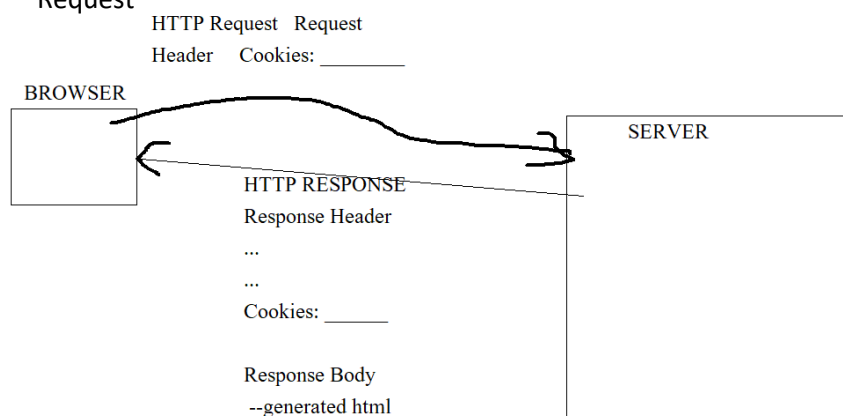
- Cookies
- URL Rewriting

Cookie = it is a small file = 1kb that holds a key=value

sessionId=76ru899

This cookie is appended to HTTP Response by the SERVER - sent to the client

When browser makes the next request to the server ---- cookie goes as a part of Http Request



## URL Rewriting

**SERVER ----->CLIENT**

Http response

Response header

Response body

<form action="/login?sessionId=76ru899" method=post>

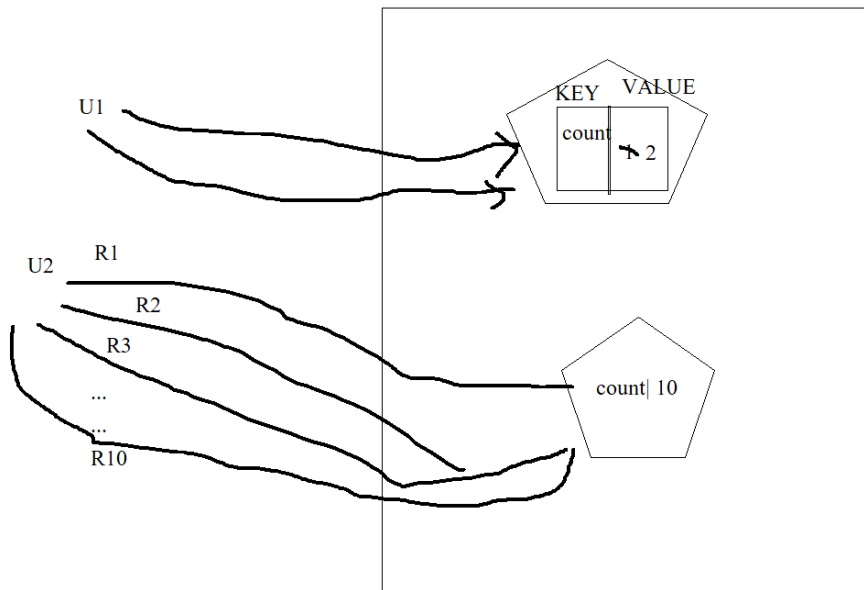
....  
....  
</form>

Client ----->Server

Client clicks on the submit button ---the url is sent to server

url : <http://localhost:8080/APPNAME/login?sessionId=76ru899>

EXAMPLE -----



javax.servlet.http.HttpSession **interface** !!! The implementation is done by tomcat container !!!

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

getSession API ----- returns the Session object

If the request is coming with a session id then it will return the  
EXISTING session object

IF the request is NOT coming a session id then it will return a NEW  
Session object

HW -----

Write the HitServlet that shows user wise hit count and global hit count .

Add Logout functionality

Run it on different browsers !!

-----  
-----  
-----

JSP ----- Java Server Pages

**Servlet - 1. dynamic html generation , 2. data processing DB access , 3. Session Management**

Servlet had some drawbacks -----

1. It is very complicated to write COMPLEX html with CSS and java script inside the servlet  
In other words --- it is COMPLEX to EMBED html in Java class!!!
2. Generating VIEW intense applications through servlet is not good idea.

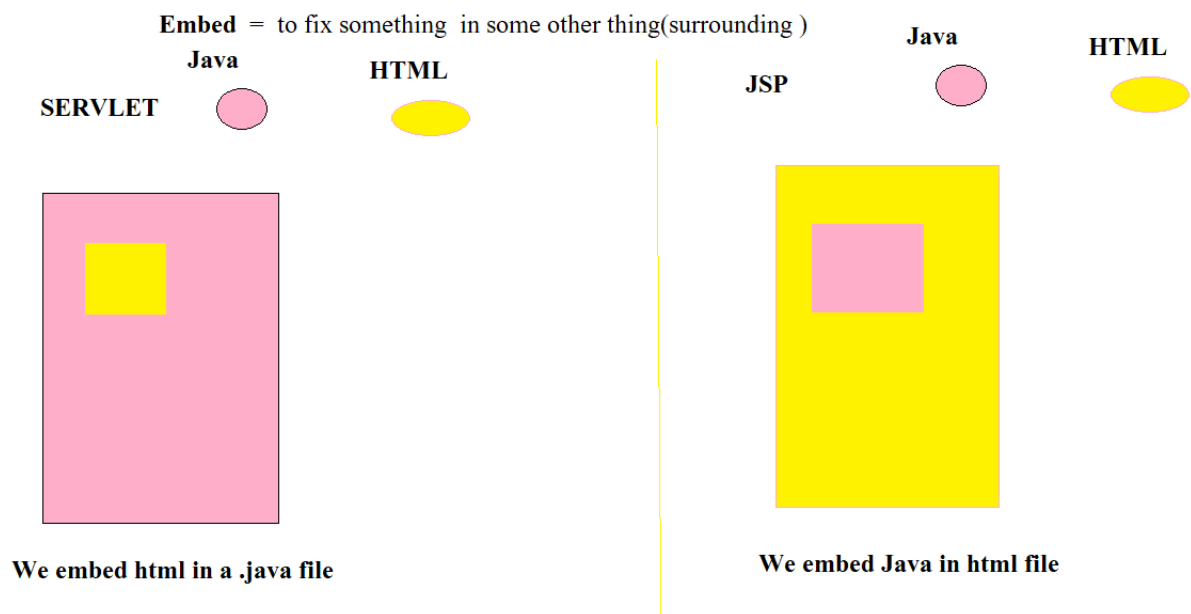
VIEW = HTML

VIEW intense = LOT of html , css and javascript !!!

TO overcome this drawback ---- JSP was introduced !!!

JSP --- we embed Java in HTML !!!

JSP ---helps in dynamic html generation just like servlet ----but it is easier to write VIEW intensive programs



JSP

JSP is translated to a SERVLET } } } } Final product on SERVER SIDE is the SERVLET !!!

Ultimately html is generated by the SERVLET !!!!

Only we don't write the servlet !!!

We write JSP !!

JSP is translated to SERVLET by a **JSP compiler** inside tomcat !!!

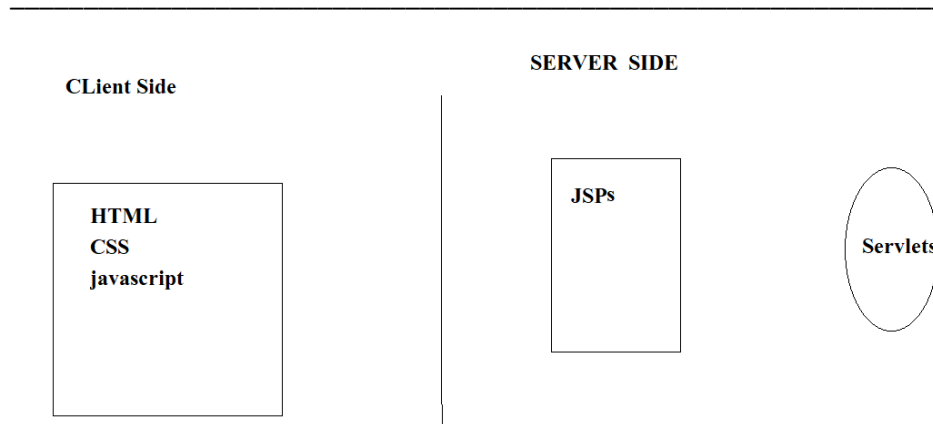
JSP compiler is called as CATALINA in tomcat = **JSP ENGINE**

**JSP ----->JSP ENGINE ----->SERVLET**

Design ----- One application may have both

We write JSP ---for VIEW

We write Servlet --- for DB access/data processing



**JSP and Servlets = SERVER SIDE WEB Components !!!! They never go to client side  
ONLY the htmls ,CSS , javascript generated by these web components go to client side !!!**

---

Servlet

<http://localhost:8080/JSPExamples/first> } /first is the URL pattern

JSP

<http://localhost:8080/JSPExamples/first.jsp> } /first.jsp is the JSP name

---

Ex1 ----- Show hello 100 times !!!

SERVLET CODE THAT WE WROTE -----

```
pw.append("<html><body>");
    for(int i=0;i<num;i++) {
        pw.append("<p>"+i+" Hello</p>");
    }
    pw.append("</body></html>");
```

---

JSP CODE

```
<html>
<body>
    <scriptlet = Java code that is embedded in html >
    <p> <output expression to show SINGLE value of a variable > Hello </p>
    <scriptlet>
</body>
</html>
```

---

```
<html>
<body>
  <% for( int i=0;i<100;i++) { %>
    <p> <%= i %>Hello </p>
  <% } %>
</body>
</html>
```

---

Ex 2 - Accept number from user and show hello those many times !!!

Way 1 ----- Create two pages one for input and one hello

Way 2 -----Do everything in one page

JSP has a list of implicit objects == objects that are already created === readymade

1. request
2. response
3. session
4. application ===== similar to ServletContext object
5. out =====similar to pw that we create in servlet

Ex3 --- write a JSP page to insert a row in the book table as discussed in class !!!

---

Tomorrow -----

DAO design to add DB code in classes !!  
Requestdispatcher , Session Management example in JSP  
JSTL tags

---

