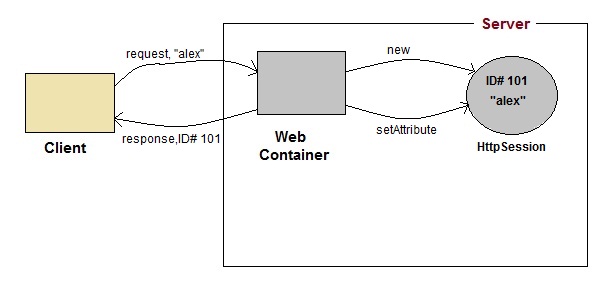
**What is HttpSession?**

**HttpSession** object is used to store entire session with a specific client. We can store, retrieve and remove attribute from **HttpSession** object. Any servlet can have access to **HttpSession** object throughout the getSession() method of the **HttpServletRequest** object.

**How HttpSession works**



1. On client's first request, the **Web Container** generates a unique session ID and gives it back to the client with response. This is a temporary session created by web container.
2. The client sends back the session ID with each request. Making it easier for the web container to identify where the request is coming from.
3. The **Web Container** uses this ID, finds the matching session with the ID and associates the session with the request.

**HttpSession Interface**



**Some Important Methods of HttpSession**

|  |  |
| --- | --- |
| **Methods** | **Description** |
| long getCreationTime() | returns the time when the session was created, measured in milliseconds since midnight January 1, 1970 GMT. |
| String getId() | returns a string containing the unique identifier assigned to the session. |
| long getLastAccessedTime() | returns the last time the client sent a request associated with the session |
| int getMaxInactiveInterval() | returns the maximum time interval, in seconds. |
| void invalidate() | destroy the session |
| boolean isNew() | returns true if the session is new else false |
| void setMaxInactiveInterval(int interval) | Specifies the time, in seconds,after servlet container will invalidate the session. |

**Complete Example demonstrating usage of HttpSession**

All the files mentioned below are required for the example.

**index.html**

<form method="post" action="**Validate**">

User: <input type="text" name="user" /><br/>

Password: <input type="text" name="pass" ><br/>

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

</form>

**web.xml**

<web-app..>

<servlet>

<servlet-name>**Validate**</servlet-name>

<servlet-class>*Validate*</servlet-class>

</servlet>

<servlet>

<servlet-name>**Welcome**</servlet-name>

<servlet-class>*Welcome*</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Validate</servlet-name>

<url-pattern>/Validate</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Welcome</servlet-name>

<url-pattern>/Welcome</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>**index.html**</welcome-file>

</welcome-file-list>

</web-app>

**Validate.java**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class *Validate* **extends** HttpServlet {

protected void **doPost**(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

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

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

if(pass.*equals*("1234"))

{

//creating a session

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

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

response.**sendRedirect**("Welcome");

}

}

}

**Welcome.java**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class *Welcome* **extends** HttpServlet {

protected void **doGet**(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

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

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

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

out.println("Hello "+**user**);

}

}

**Using Cookies for Session Management**

**Cookies** are small pieces of information that are sent in response from the web server to the client. **Cookies** are the simplest technique used for storing client state.

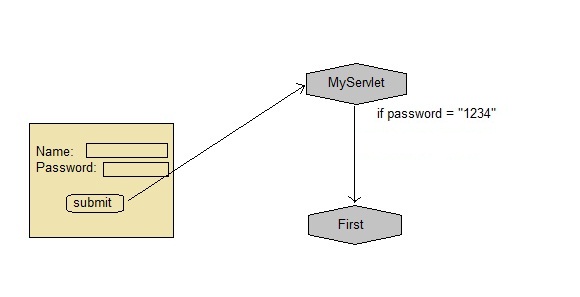
**Cookies** are stored on client's computer. They have a lifespan and are destroyed by the client browser at the end of that lifespan.

Using Cookies for storing client state has one shortcoming though, if the client has turned of COokie saving settings in his browser then, client state can never be saved because the browser will not allow the application to store cookies.

**Cookies API**

Cookies are created using **Cookie** class present in Servlet API. Cookies are added to **response** object using the addCookie() method. This method sends cookie information over the HTTP response stream. getCookies() method is used to access the cookies that are added to response object.

**Example demonstrating usage of Cookies**



Below mentioned files are required for the example:

**index.html**

<form method="post" action="**validate**">

Name:<input type="text" name="user" /><br/>

Password:<input type="text" name="pass" ><br/>

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

</form>

**web.xml**

<web-app...>

<servlet>

<servlet-name>**validate**</servlet-name>

<servlet-class>**MyServlet**</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>**validate**</servlet-name>

<url-pattern>/validate</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>**First**</servlet-name>

<servlet-class>**First**</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>**First**</servlet-name>

<url-pattern>/First</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>**index.html**</welcome-file>

</welcome-file-list>

</web-app>

**MyServlet.java**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class *MyServlet* **extends** HttpServlet {

protected void **doPost**(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

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

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

if(pass.*equals*("1234"))

{

Cookie **ck** = new Cookie("*username*",*name*);

response.**addCookie**(*ck*);

response.sendRedirect("*First*");

}

}

}

**First.java**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class *First* **extends** HttpServlet {

protected void **doGet**(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

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

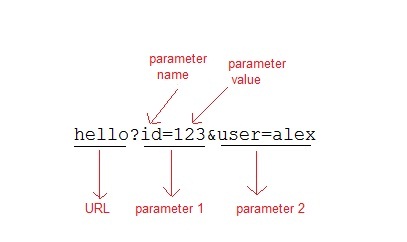
out.println("Welcome "+**cks[0].getValue()**);

}

}

In URL rewriting, a token(parameter) is added at the end of the URL. The token consist of name/value pair seperated by an equal(=) sign.

**For Example:**



When the User clicks on the URL having parameters, the request goes to the **Web Container** with extra bit of information at the end of URL. The **Web Container** will fetch the extra part of the requested URL and use it for session management.

The getParameter() method is used to get the parameter value at the server side.

**Example demonstrating usage of URL rewriting**

Below mentioned files are required for the example:

**index.html**

<form method="post" action="**validate**">

Name:<input type="text" name="user" /><br/>

Password:<input type="text" name="pass" ><br/>

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

</form>

**web.xml**

<web-app...>

<servlet>

<servlet-name>**validate**</servlet-name>

<servlet-class>**MyServlet**</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>**validate**</servlet-name>

<url-pattern>/validate</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>**First**</servlet-name>

<servlet-class>**First**</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>**First**</servlet-name>

<url-pattern>/First</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>**index.html**</welcome-file>

</welcome-file-list>

</web-app>

**MyServlet.java**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class *MyServlet* **extends** HttpServlet {

protected void **doPost**(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

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

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

if(pass.*equals*("1234"))

{

response.sendRedirect(*"First?user\_name="+****name****+""*);

}

}

}

**First.java**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class *First* **extends** HttpServlet {

protected void **doGet**(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

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

out.println("Welcome "+user);

}

}