

SERVLET and JSP

1. Understanding Session Concept
2. Session implementation in project
3. Fetching value of Session in multiple jsp pages

1. *Understanding Session Concept*

Session is a period of time in which user sends multiple requests and receives multiple response.

Request object which we used in jsp and servlet is stateless means it is not retaining data if we move from one page to another page and once gain to next page.

To move data at multiple places we need to use session object.

How to get session object?

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

http session object will be created by the container when you call getSession () method first time.

```
HttpSession session = request.getSession (). // This line generates one id internally which is called as jsessionid
```

In this concept we need to identify old session and new session.

- **Old session:**
 - If jsession id sent from client is same as that of stored on server then that is called as old session.
- **New session:**
 - If jsession id sent from client is not same as that of stored on server then session id is generated on server will be new session.

There are 2 ways to write session statement:

1. `HttpSession session = request.getSession(true);`
 - a. If you write this in servlet then first it checks for old session if not it will generate new session id.
 - b. If true is not written, then by default it is true unless you mention false.
2. `HttpSession session = request.getSession(false);`
 - a. If you write this in servlet then first it checks for old session and returns same id otherwise it will not generate new session, but it returns null.

Use of below line in details.

HttpSession session =request.getSession().

Checks whether the incoming request contains the cookies **with name JsessionId or not.**

If the incoming request contains the cookies with the name JSESSIONID then collect the value of the cookies which is JsessionId and checks whether that matches with server's JsessionId if matches then it picks session object and identify as a old client.

If the incoming request does not contain the cookie with name JSESSIONID then following steps will happen.

1. Create session object.
2. Generate unique SESSION ID.
3. Stores the session ID in session object.
4. Create the cookies with the name JSESSIONID.
5. Adds the cookies the response object.
6. Return the session object.

There are 4 session management techniques: -

1. HttpSession object – used mostly in industry.
2. Cookies – point #1 covers it.
3. URL-rewriting – not used much.
4. Hidden fields – not used much.

You can use the following to store the clients' conversational data:

1. http session
2. Cookies

2. Session implementation in project

Objective: We will be logging in to application then userid will be stored in request and session objects. We will be traversing to various pages to check if value of userid is persisted on other pages. also on last page will have logout button.

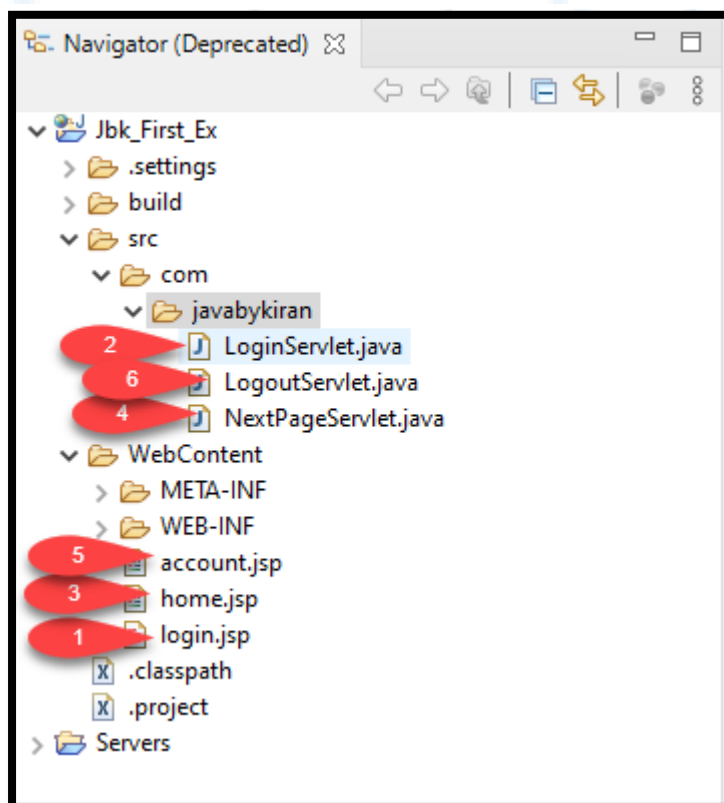
Before implementing project, we need to know below questions and answers.

1. How to check if session is new or old.
 - a. session.isNew()
2. How to invalidate or expire a session.
 - a. Session.invalidate()
3. How to print session id.
 - a. Session.getId()
4. How to create a session.
 - a. HttpSession session=request.getSession().

We need below files from old project which we already developed.

- 1) Login.jsp
- 2) Home.jsp
- 3) Account.jsp
- 4) LoginServlet.java
- 5) NextPageServlet.java
- 6) LogoutServlet.java

Project structure with data flow is shown below. Flow will be in sequence.



Below is code we need to follow.

Login.jsp

```
<h2>JAVA BY KIRAN</h2>
<h2>Account Login</h2>

<%
String message = (String) request.getAttribute("msg");
if(message!=null){
    out.print(message);
}
%>

<form action="login" method="post" />
<table>
    <tr>
        <td>Username:</td>
        <td><input type="text" name="uname" /></td>
    </tr>
    <tr>
        <td>Password:</td>
        <td><input type="text" name="pword" /></td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" value="Login" />
    </tr>
</table>
</form>
```

LoginServlet.java

```
package com.javabykiran;

import java.io.IOException;
import java.util.ArrayList;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/login")
public class LoginServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        HttpSession session = request.getSession();
        // printing id just for checking
        System.out.println("session Id > " + session.getId());
        // checking if session is new
        System.out.println("session status> " + session.isNew());

        String uname = request.getParameter("uname");
        // now we will be setting this in request and session.

        // setting in request
        request.setAttribute("unameInRequest", uname);
        // setting in session
        session.setAttribute("unameInRequest", uname);

        // Navigating to home.jsp
        RequestDispatcher rd = request.getRequestDispatcher("home.jsp");
        rd.forward(request, response);
    }
}
```

Home.jsp

```
<% @page import="java.util.ArrayList"%>
<%
String unameReq = (String) request.getAttribute("unameInRequest");
out.println("unameInRequest >> " + unameReq);
String unameSes = (String) session.getAttribute("unameInSession");
out.println("unameInSession >> " + unameSes);
%>
<form action="NextPageServlet">
    <input type="submit" value="submit" />
```

```
</form>
```

NextPageServlet.java

```
package com.javabykiran;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class NextPageServlet
 */
@WebServlet("/NextPageServlet")
public class NextPageServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // Navigating to home.jsp
        RequestDispatcher rd = request.getRequestDispatcher("account.jsp");
        rd.forward(request, response);

    }

}
```

Account.jsp

```
<% @page import="java.util.ArrayList"%>
<%
String unameReq = (String) request.getAttribute("unameInRequest");
out.println("checking unameInRequest >> " + unameReq);
String unameSes = (String) session.getAttribute("unameInSession");
out.println("checking unameInSession >> " + unameSes);
%>
<form action="LogoutServlet">
    <input type="submit" value="Logout" />
```

```
</form>
```

logoutServlet.java

```
package com.javabykiran;

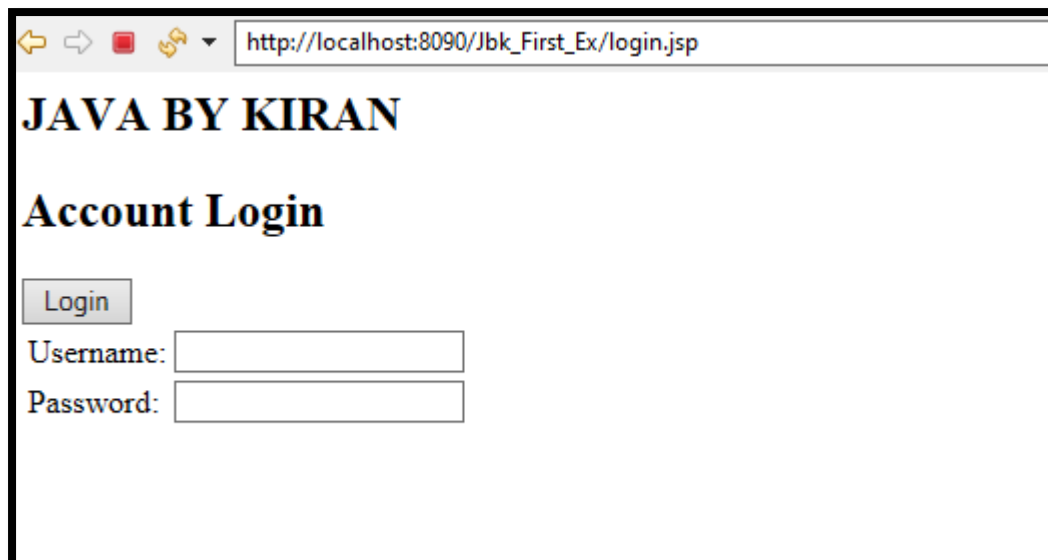
import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
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 {

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        HttpSession session = request.getSession();
        // printing id just for checking
        System.out.println("session Id > " + session.getId());
        // checking if session is new
        System.out.println("session status> " + session.isNew());
        //invalidating session
        session.invalidate();
        // Navigating to home.jsp
        request.setAttribute("msg", "session is expired...");
        RequestDispatcher rd = request.getRequestDispatcher("login.jsp");
        rd.forward(request, response);
    }
}
```

Let's run a project and observe.



http://localhost:8090/Jbk_First_Ex/login.jsp

JAVA BY KIRAN

Account Login

Login

Username:

Password:



http://localhost:8090/Jbk_First_Ex/login

unameInRequest >> Javabykiran

unameInSession >> Javabykiran

submit

We are able to print data from session and request as well.

Console

tomcat v8.5 Server at localhost [Apache Tomcat/8.5.20] C:\Program Files\Java\jdk-11.0.9\bin\java.exe (13-Jul-2021)

session Id > 41B5EA5B73D056317B4EF63D482AAFCB

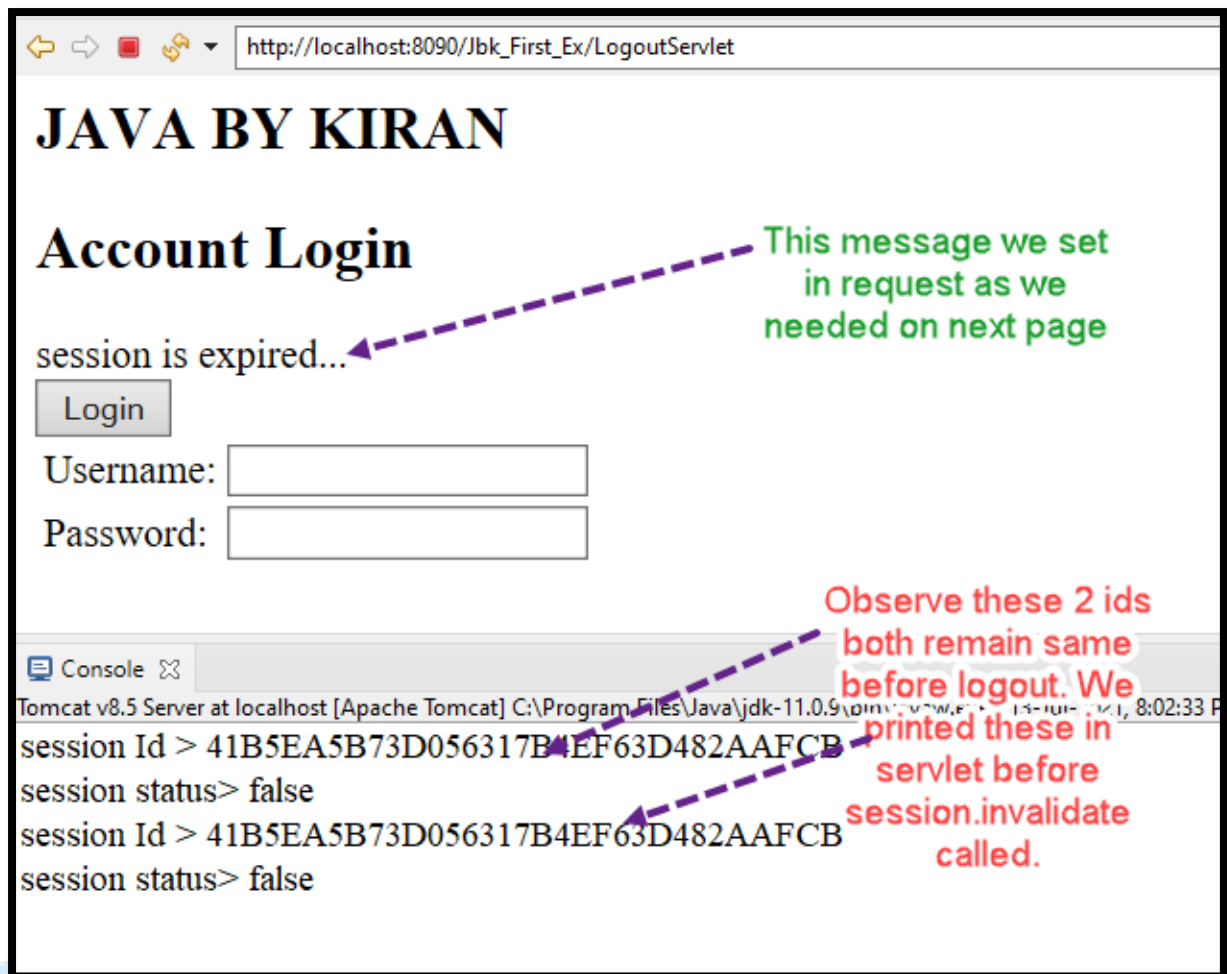
session status > false

This id is automatically generated

The screenshot shows a web browser window with the address bar displaying `http://localhost:8090/Jbk_First_Ex/NextPageServlet`. The page content includes the text `unameInRequest >> null` and `unameInSession >> Javabykiran`. A "Logout" button is visible below the first text. Two purple ovals highlight the values `null` and `Javabykiran`. Dashed purple arrows point from these ovals to green text annotations below. The annotation for `null` states: "data from request lost as we move to other pages because **request** object is stateless". The annotation for `Javabykiran` states: "data from session not lost as we move to other pages because **session** object is statefull". At the bottom, a console window shows the following output:

```
Tomcat v8.5 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-11.0.9\bin\javaw.exe (13-Jul-2021, 8:02:33 PM)
session Id > 41B5EA5B73D056317B4EF63D482AAFCB
session status> false
```

javabykiran
java | selenium | python



3. Fetching value of Session in multiple jsp pages

In above example we have fetches session values in multiple pages to name account.jsp and home.jsp

Homework:

1. Do same thing for password.
2. Show welcome message on every page “welcome kiran” like this.
3. Change logout message.
4. Add one more page after account.jsp as myaccount.jsp flow should not be changed.

Download:

https://drive.google.com/drive/folders/1TKap6vODBcgKsbHATSaxLbfn4OX_E2Cu?usp=sharing

javabyKiran
java | selenium | python