

Session-9

Session Management

Introduction:

WHAT IS SESSION

The following are the techniques used in session tracking process

- a. Cookies
- b. Hidden form fields
- c. URL re-writing
- d. Http Session

1)Cookie session tracking process

Define cookie?

The small piece of information persisted between multiple client requests is known as cookie

The following are the important fields of cookie:

1.name

2. value

3. optional attributes

 |--path

 |--comment

4. domain qualifiers

 |-age

 |-version

Define cookie class:

Cookie is a class , using this class we will perform tracking

Example:

Javap javax.servlet.http.cookie

Explanation:

1. cookie is a class coming from javax.servlet.http package of servlet
2. cookie class is implemented from java.lang.clonable interface
3. cookie class is also implemented from java.io.serializable interface
4. make sure a constructor is available for this cookie class
5. this constructor will have two parameters (parameterized constructor)

Example:

```
public javax.servlet.http.Cookie(java.lang.String.java.lang.String);
```

Methods used in cookies:

Two methods will be used in cookies they are

1. Public void addCookie(Cookie ck)
2. Public Cookie[] get_cookies():

Syntax for creating cookie:

```
Cookie ck=new Cookie("name","value"); //creating cookie object
```

```
response.addCookie(ck) //adding cookie in the response
```

syntax for deleting cookie:

```
Cookie ck=new Cookie("name",""); //deleting value of cookie
```

```
Ck.setMaxAge(0); //changing the maximum age to 0 seconds
```

```
Response.addCookie(ck); //adding cookie in the response
```

Syntax for getting cookies:

```
Cookie ck[]=request.getCookies();
```

Cookie Demonstration program:

index.html

```
<form action="servlet1" method="post">  
Name:<input type="text" name="userName"/><br/>  
<input type="submit" value="submit"/>  
</form>
```

FirstServlet.java

```
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
public class FirstServlet extends HttpServlet {  
    public void doPost(HttpServletRequest request, HttpServletResponse response){  
        try{  
            response.setContentType("text/html");  
            PrintWriter out = response.getWriter();  
  
            String n=request.getParameter("userName");  
            out.print("Welcome "+n);
```

```
Cookie ck=new Cookie("uname",n);//creating cookie object  
response.addCookie(ck);//adding cookie in the response
```

```
//creating submit button
```

```
out.print("<form action='servlet2'>");
```

```
out.print("<input type='submit' value='go'>");
```

```
out.print("</form>");
```

```
out.close();
```

```
    }catch(Exception e){System.out.println(e);}
}
```

```
}
```

SecondServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SecondServlet extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response){
        try{
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();

            Cookie ck[]=request.getCookies();
            out.print("Hello "+ck[0].getValue());

            out.close();

        }catch(Exception e){System.out.println(e);}
    }
}
```

web.xml

```
<web-app>
<servlet>
<servlet-name>s1</servlet-name>
<servlet-class>FirstServlet</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>s1</servlet-name>
<url-pattern>/servlet1</url-pattern>
</servlet-mapping>
<servlet>
<servlet-name>s2</servlet-name>
<servlet-class>SecondServlet</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>s2</servlet-name>
<url-pattern>/servlet2</url-pattern>
</servlet-mapping>

</web-app>
```

Servlet Login and Logout Example using Cookies

In this application, we have created following files.

1. index.html
2. link.html
3. login.html
4. LoginServlet.java
5. LogoutServlet.java
6. ProfileServlet.java
7. web.xml

File: index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Servlet Login Example</title>
</head>
<body>

<h1>Welcome to Login App by Cookie</h1>
<a href="login.html">Login</a> |
<a href="LogoutServlet">Logout</a> |
<a href="ProfileServlet">Profile</a>

</body>
</html>
```

File: link.html

```
<a href="login.html">Login</a> |
<a href="LogoutServlet">Logout</a> |
<a href="ProfileServlet">Profile</a>
<hr>
```

File: login.html

```
<form action="LoginServlet" method="post">
Name:<input type="text" name="name"><br>
Password:<input type="password" name="password"><br>
<input type="submit" value="login">
</form>
```

File: **LoginServlet.java**

```
package com.javatpoint;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class LoginServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();

        request.getRequestDispatcher("link.html").include(request, response);

        String name=request.getParameter("name");
        String password=request.getParameter("password");

        if(password.equals("admin123")){
            out.print("You are successfully logged in!");
            out.print("<br>Welcome, "+name);

            Cookie ck=new Cookie("name",name);
            response.addCookie(ck);
        }else{
            out.print("sorry, username or password error!");
            request.getRequestDispatcher("login.html").include(request, response);
        }

        out.close();
    }
}
```


File: **LogoutServlet.java**

```
package com.javatpoint;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class LogoutServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();

        request.getRequestDispatcher("link.html").include(request, response);

        Cookie ck=new Cookie("name","");
        ck.setMaxAge(0);
        response.addCookie(ck);

        out.print("you are successfully logged out!");
    }
}
```

File: **ProfileServlet.java**

```
package com.javatpoint;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class ProfileServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();

        request.getRequestDispatcher("link.html").include(request, response);

        Cookie ck[]=request.getCookies();
        if(ck!=null){
            String name=ck[0].getValue();
            if(!name.equals("") || name!=null){
                out.print("<b>Welcome to Profile</b>");
                out.print("<br>Welcome, "+name);
            }
        }else{
            out.print("Please login first");
            request.getRequestDispatcher("login.html").include(request, response);
        }
        out.close();
    }
}
```

File: **web.xml**

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID"
version="2.5">

    <servlet>
        <description></description>
        <display-name>LoginServlet</display-name>
        <servlet-name>LoginServlet</servlet-name>
        <servlet-class>com.javatpoint.LoginServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>LoginServlet</servlet-name>
        <url-pattern>/LoginServlet</url-pattern>
    </servlet-mapping>

    <servlet>
        <description></description>
        <display-name>ProfileServlet</display-name>
        <servlet-name>ProfileServlet</servlet-name>
        <servlet-class>com.javatpoint.ProfileServlet</servlet-class>
```

```
</servlet>
<servlet-mapping>
<servlet-name>ProfileServlet</servlet-name>
<url-pattern>/ProfileServlet</url-pattern>
</servlet-mapping>
<servlet>
<description></description>
<display-name>LogoutServlet</display-name>
<servlet-name>LogoutServlet</servlet-name>
<servlet-class>com.javatpoint.LogoutServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>LogoutServlet</servlet-name>
<url-pattern>/LogoutServlet</url-pattern>
</servlet-mapping>
</web-app>
```

2) Hidden Form Field

In case of Hidden Form Field a **hidden (invisible) textfield** is used for maintaining the state of an user.

In such case, we store the information in the hidden field and get it from another servlet. This approach is better if we have to submit form in all the pages and we don't want to depend on the browser.

Let's see the code to store value in hidden field.

Syntax: `<input type="hidden" name="uname" value="Vimal Jaiswal">`

index.html

```
<form action="servlet1">  
Name:<input type="text" name="userName"/><br/>  
<input type="submit" value="go"/>  
</form>
```

FirstServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class FirstServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response){
        try{

            response.setContentType("text/html");
            PrintWriter out = response.getWriter();

            String n=request.getParameter("userName");
            out.print("Welcome "+n);

            //creating form that have invisible textfield
            out.print("<form action='servlet2'>");
            out.print("<input type='hidden' name='uname' value='"+n+"'>");
            out.print("<input type='submit' value='go'>");
            out.print("</form>");
            out.close();

            }catch(Exception e){System.out.println(e);}
        }
    }
}
```

SecondServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SecondServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        try{
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();

            //Getting the value from the hidden field
            String n=request.getParameter("uname");
            out.print("Hello "+n);

            out.close();

        }catch(Exception e){System.out.println(e);}
    }
}
```

web.xml

```
<web-app>

<servlet>

<servlet-name>s1</servlet-name>

<servlet-class>FirstServlet</servlet-class>

</servlet>


<servlet-mapping>

<servlet-name>s1</servlet-name>

<url-pattern>/servlet1</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>s2</servlet-name>

<servlet-class>SecondServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>s2</servlet-name>

<url-pattern>/servlet2</url-pattern>

</servlet-mapping>


</web-app>
```


3)URL Rewriting

In URL rewriting, we append a token or identifier to the URL of the next Servlet or the next resource. We can send parameter name/value pairs using the following format:

`url?name1=value1&name2=value2&??`

A name and a value is separated using an equal = sign, a parameter name/value pair is separated from another parameter using the ampersand(&). When the user clicks the hyperlink, the parameter name/value pairs will be passed to the server. From a Servlet, we can use `getParameter()` method to obtain a parameter value

Advantage of URL Rewriting

1. It will always work whether cookie is disabled or not (browser independent).
2. Extra form submission is not required on each pages.

Disadvantage of URL Rewriting

1. It will work only with links.
2. It can send Only textual information.

URL Rewriting Demo

index.html

```
<form action="servlet1">  
Name:<input type="text" name="userName"/><br/>  
<input type="submit" value="go"/>  
</form>
```

FirstServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class FirstServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response){
        try{

            response.setContentType("text/html");
            PrintWriter out = response.getWriter();

            String n=request.getParameter("userName");
            out.print("Welcome "+n);

            //appending the username in the query string
            out.print("<a href='servlet2?uname="+n+"'>visit</a>");

            out.close();

        }catch(Exception e){System.out.println(e);}
    }
}
```

SecondServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SecondServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        try{

            response.setContentType("text/html");
            PrintWriter out = response.getWriter();

            //getting value from the query string
            String n=request.getParameter("uname");
            out.print("Hello "+n);

            out.close();

        }catch(Exception e){System.out.println(e);}
    }
}
```

web.xml

```
<web-app>

<servlet>

<servlet-name>s1</servlet-name>

<servlet-class>FirstServlet</servlet-class>

</servlet>


<servlet-mapping>

<servlet-name>s1</servlet-name>

<url-pattern>/servlet1</url-pattern>

</servlet-mapping>


<servlet>

<servlet-name>s2</servlet-name>

<servlet-class>SecondServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>s2</servlet-name>

<url-pattern>/servlet2</url-pattern>

</servlet-mapping>

</web-app>
```

4) HttpSession interface

In such case, container creates a session id for each user. The container uses this id to identify the particular user. An object of HttpSession can be used to perform two tasks:

1. bind objects
2. view and manipulate information about a session, such as the session identifier, creation time, and last accessed time.

How to get the HttpSession object ?

The HttpServletRequest interface provides two methods to get the object of HttpSession:

1. **public HttpSession getSession():** Returns the current session associated with this request, or if the request does not have a session, creates one.
2. **public HttpSession getSession(boolean create):** Returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session.

Commonly used methods of HttpSession interface

1. **public String getId():** Returns a string containing the unique identifier value.
2. **public long getCreationTime():** Returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.
3. **public long getLastAccessedTime():** Returns the last time the client sent a request associated with this session, as the number of milliseconds since midnight January 1, 1970 GMT.
4. **public void invalidate():** Invalidates this session then unbinds any objects bound to it.

HttpSession Demo

index.html

```
<form action="servlet1">  
Name:<input type="text" name="userName"/><br/>  
<input type="submit" value="go"/>  
</form>
```

FirstServlet.java

```
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
  
public class FirstServlet extends HttpServlet {  
  
    public void doGet(HttpServletRequest request, HttpServletResponse response){  
        try{  
  
            response.setContentType("text/html");  
            PrintWriter out = response.getWriter();  
  
            String n=request.getParameter("userName");  
            out.print("Welcome "+n);  
  
            HttpSession session=request.getSession();  
            session.setAttribute("uname",n);  
  
            out.print("<a href='servlet2'>visit</a>");  
  
            out.close();  
  
        }catch(Exception e){System.out.println(e);}  
    } }
```

SecondServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SecondServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        try{

            response.setContentType("text/html");
            PrintWriter out = response.getWriter();

            HttpSession session=request.getSession(false);
            String n=(String)session.getAttribute("uname");
            out.print("Hello "+n);

            out.close();

        }catch(Exception e){System.out.println(e);}
    } }
```

web.xml

```
<web-app>
<servlet>
<servlet-name>s1</servlet-name>
<servlet-class>FirstServlet</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>s1</servlet-name>
<url-pattern>/servlet1</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>s2</servlet-name>
<servlet-class>SecondServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>s2</servlet-name>
<url-pattern>/servlet2</url-pattern>
</servlet-mapping>
</web-app>
```


Servlet HttpSession Login and Logout Example

We can bind the objects on HttpSession instance and get the objects by using `setAttribute` and `getAttribute` methods.

In the previous page, we have learnt about what is HttpSession, How to store and get data from session object etc.

Here, we are going to create a real world login and logout application without using database code. We are assuming that password is admin123

In this application, we have created following files.

1. index.html
2. link.html
3. login.html
4. LoginServlet.java
5. LogoutServlet.java
6. ProfileServlet.java
7. web.xml

File: index.html

```
<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Servlet Login Example</title>

</head>

<body>
```

```
<h1>Login App using HttpSession</h1>
<a href="login.html">Login</a>|
<a href="LogoutServlet">Logout</a>|
<a href="ProfileServlet">Profile</a>

</body>
</html>
```

File: link.html

```
<a href="login.html">Login</a> |
<a href="LogoutServlet">Logout</a> |
<a href="ProfileServlet">Profile</a>
<hr>
File: login.html
<form action="LoginServlet" method="post">
Name:<input type="text" name="name"><br>
Password:<input type="password" name="password"><br>
<input type="submit" value="login">
</form>
```

File: LoginServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class LoginServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        request.getRequestDispatcher("link.html").include(request, response);

        String name=request.getParameter("name");
        String password=request.getParameter("password");

        if(password.equals("admin123")){
            out.print("Welcome, "+name);
            HttpSession session=request.getSession();
            session.setAttribute("name",name);
        }
        else{
            out.print("Sorry, username or password error!");
            request.getRequestDispatcher("login.html").include(request, response);
        }
        out.close();
    }
}
```

File: LogoutServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class LogoutServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();

        request.getRequestDispatcher("link.html").include(request,
response);

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

        out.print("You are successfully logged out!");

        out.close();
    }
}
```

File: ProfileServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class ProfileServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        request.getRequestDispatcher("link.html").include(request,
response);

        HttpSession session=request.getSession(false);
        if(session!=null){
            String name=(String)session.getAttribute("name");

            out.print("Hello, "+name+" Welcome to Profile");
        }
        else{
            out.print("Please login first");
            request.getRequestDispatcher("login.html").include(request,
response);
        }
        out.close();
    }
}
```

File: web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID"
version="2.5">

    <servlet>

    <description></description>

    <display-name>LoginServlet</display-name>

    <servlet-name>LoginServlet</servlet-name>

    <servlet-class>LoginServlet</servlet-class>

    </servlet>

    <servlet-mapping>

    <servlet-name>LoginServlet</servlet-name>

    <url-pattern>/LoginServlet</url-pattern>

    </servlet-mapping>

    <servlet>

    <description></description>

    <display-name>ProfileServlet</display-name>

    <servlet-name>ProfileServlet</servlet-name>

    <servlet-class>ProfileServlet</servlet-class>
```

```
</servlet>
<servlet-mapping>
<servlet-name>ProfileServlet</servlet-name>
<url-pattern>/ProfileServlet</url-pattern>
</servlet-mapping>
<servlet>
<description></description>
<display-name>LogoutServlet</display-name>
<servlet-name>LogoutServlet</servlet-name>
<servlet-class>LogoutServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>LogoutServlet</servlet-name>
<url-pattern>/LogoutServlet</url-pattern>
</servlet-mapping>
</web-app>
```