Name-Shubhankar Joshi

Roll no - 18

Div- ICB

Batch – B1

Session Tracking Methods in Servlets

INTRODUCTION

Session Tracking is a way to maintain state (data) of an user. Session tracking is mechanism of tracking the client provided data and making it available to the next request from the same client. And this process is continued until the user choose to Logout or terminate the session.

Http protocol is a stateless so we need to maintain state using session tracking techniques. Each time user requests the server the server treats the request as the new request. So we need to maintain the state of an user to recognize to particular user. Session tracking is a mechanism that servlets use to maintain state about a series of requests from the same user (that is, requests originating from same browser).

Session Tracking methods:-

1. User Authorization
2. Cookies
3. URL Rewriting
4. Hidden field

Cookie

Cookies are the mostly used technology for session tracking. Cookie is a key value pair of information, sent by the server to the browser. This should be saved by the browser in its space in the client computer. Whenever the browser sends a request to that server it sends the cookie along with it. Then the server can identify the client using the cookie. Session tracking is easy to implement and maintain using the cookies. Disadvantage is that, the users can opt to disable cookies using their browser.

Code-

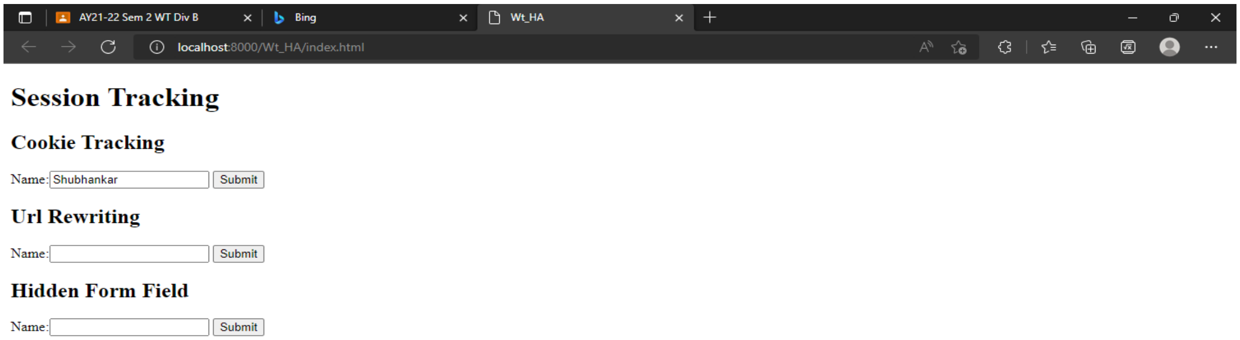
FirstServlet.java

```
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 FirstServlet extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    public void init() throws ServletException
    {
        System.out.println("-----------------------------------------");
        System.out.println(" Init method is called in "
                + this.getClass().getName());
        System.out.println("--------------------------------------");
    }

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

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

        Cookie cookie = new Cookie("userName", userName);// creating cookie
object
        response.addCookie(cookie);// 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();

    }

    public void destroy()
    {
        System.out.println("-----------------------------------------");
        System.out.println(" destroy method is called in "
                + this.getClass().getName());
        System.out.println("-----------------------------------------");
    }

}
```

SecondServlet.java

```
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 SecondServlet extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    public void init() throws ServletException
    {
        System.out.println("--------------------------------------------");
        System.out.println(" Init method is called in "
                + this.getClass().getName());
        System.out.println("---------------------------------------------");
    }

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

        Cookie cookiesArray[] = request.getCookies();

        for( Cookie cookie : cookiesArray )
        {
            out.print("Hello " + cookie.getValue());
        }

        out.close();
    }

    public void destroy()
    {
        System.out.println("---------------------------------------------");
        System.out.println(" destroy method is called in "
                + this.getClass().getName());
        System.out.println("---------------------------------------------");
    }

}
```

OUTPUT-



Session Tracking

Cookie Tracking

Name: Shubhankar Submit

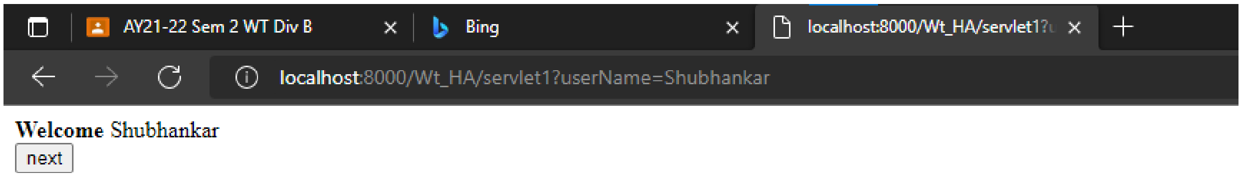
Url Rewriting

Name: Submit

Hidden Form Field

Name: Submit

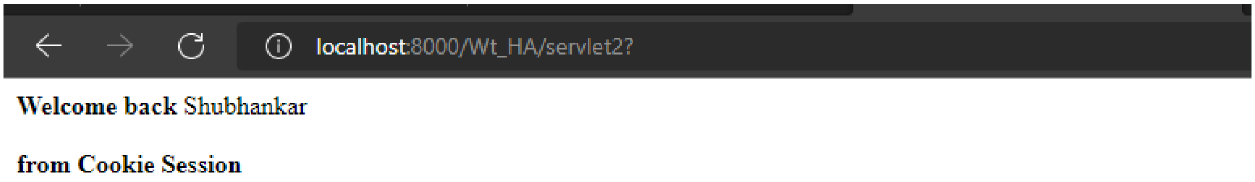
FirstServlet


AY21-22 Sem 2 WT Div B
Bing
localhost:8000/Wt_HA/servlet1?u
localhost:8000/Wt_HA/servlet1?userName=Shubhankar

Welcome Shubhankar

next

SecondServlet



Welcome back Shubhankar

from Cookie Session

URL Rewriting

In URL Rewriting we append token or identifier to the URL of the next resource.

When a request is made, additional parameter is appended with the URL. In general added additional parameter will be sessionid or sometimes the userid. It will suffice to track the session. This type of session tracking doesn't need any special support from the browser, Disadvantage is, implementing this type of session tracking is tedious. We need to keep track of the parameter as a chain link until the conversation completes and also should make sure that, the parameter doesn't clash with other application

Code:-

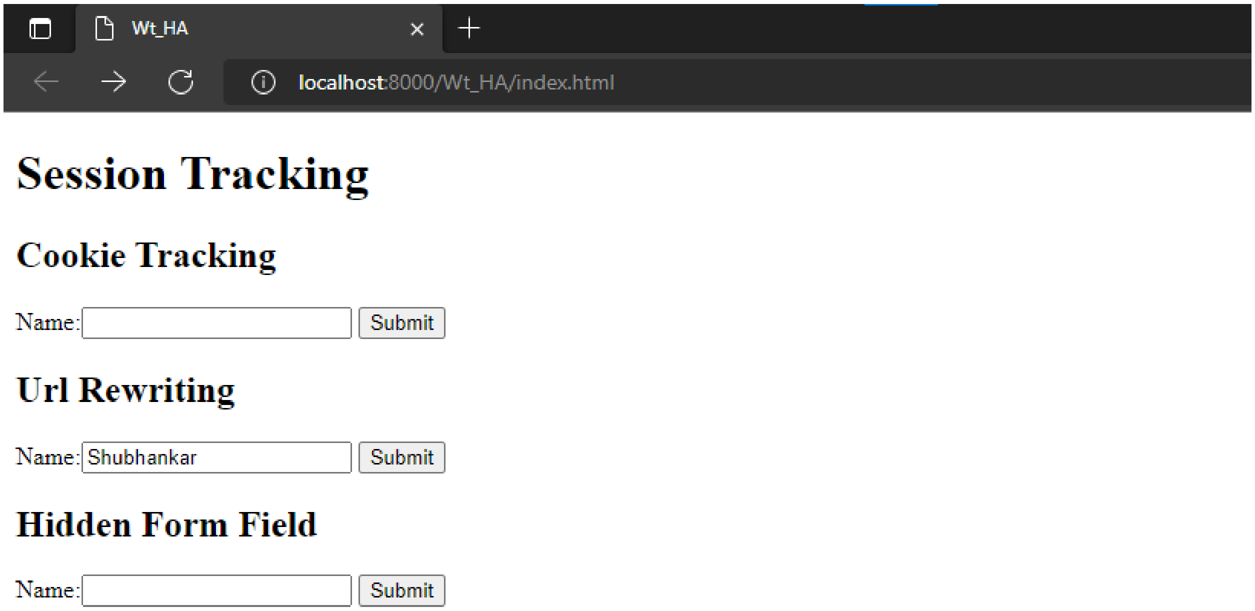
FirstServlet.java

```
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;

public class FirstServlet extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    public void init() throws ServletException
    {
        System.out.println("-----------------------------------------");
        System.out.println(" Init method is called in "
                + this.getClass().getName());
        System.out.println("--------------------------------------");
    }

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

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

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

        out.close();
    }

    public void destroy()
    {
        System.out.println("-----------------------------------------");
        System.out.println(" destroy method is called in "
                + this.getClass().getName());
        System.out.println("-----------------------------------------");
    }

}
```

SecondServlet.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;

public class SecondServlet extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    public void init() throws ServletException
    {
        System.out.println("-----------------------------------------");
        System.out.println(" Init method is called in "
                + this.getClass().getName());
        System.out.println("--------------------------------------");
    }

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

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

        out.close();

    }

    public void destroy()
    {
        System.out.println("-----------------------------------------");
        System.out.println(" destroy method is called in "
                + this.getClass().getName());
        System.out.println("-----------------------------------------");
    }

}
```

Output:-



Session Tracking

Cookie Tracking

Name: [   ] Submit

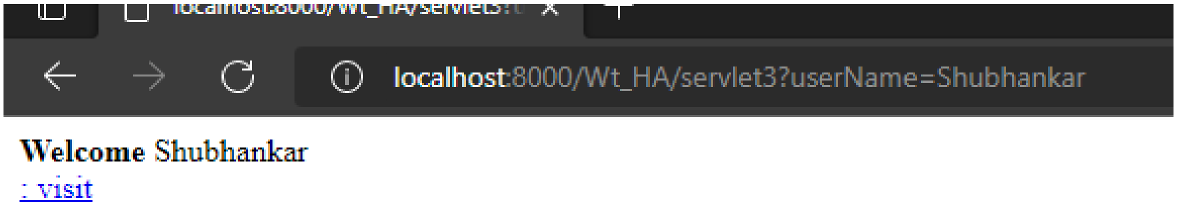
Url Rewriting

Name: [Shubhankar] Submit

Hidden Form Field

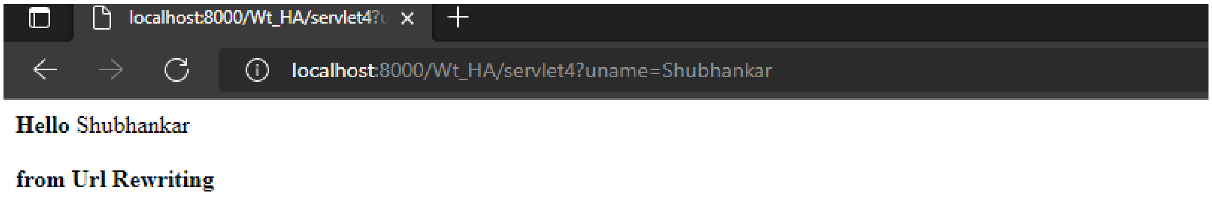
Name: [   ] Submit

FirstServlet.java


localhost:8000/Wt_HA/servlet3?u
localhost:8000/Wt_HA/servlet3?userName=Shubhankar

Welcome Shubhankar
: visit

SecondServlet.java


localhost:8000/Wt_HA/servlet4?u
localhost:8000/Wt_HA/servlet4?uname=Shubhankar

Hello Shubhankar

from Url Rewriting

HiddenForm Field

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

Code-

FirstServlet.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;

public class FirstServlet extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    public void init() throws ServletException
    {
        System.out.println("-----------------------------------------");
        System.out.println(" Init method is called in "
                + this.getClass().getName());
        System.out.println("--------------------------------------");
    }

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

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

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

    public void destroy()
    {
        System.out.println("-----------------------------------------");
        System.out.println(" destroy method is called in "
                + this.getClass().getName());
        System.out.println("-----------------------------------------");
    }

}
```

SecondServlet.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;

public class SecondServlet extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    public void init() throws ServletException
    {
        System.out.println("-----------------------------------------");
        System.out.println(" Init method is called in "
                + this.getClass().getName());
        System.out.println("--------------------------------------");
    }

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

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

    }

    public void destroy()
    {
        System.out.println("-----------------------------------------");
        System.out.println(" destroy method is called in "
                + this.getClass().getName());
        System.out.println("-----------------------------------------");
    }

}
```

WEB.XML

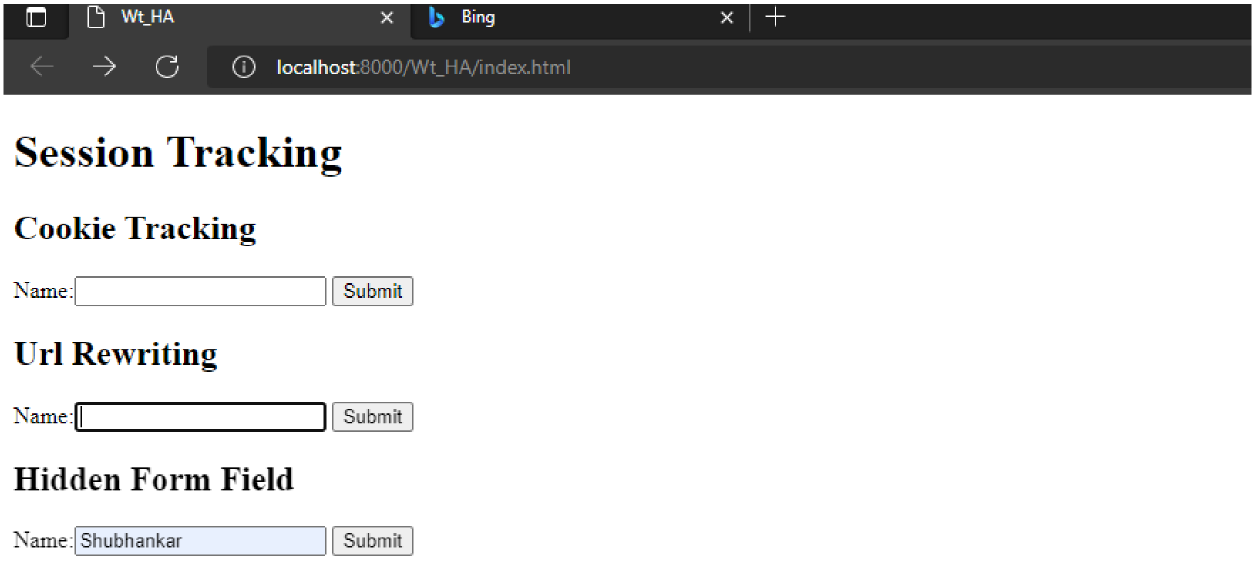
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">
```

```
  <display-name>Wt_HA</display-name>
  <servlet>
    <servlet-name>s1</servlet-name>
    <servlet-class>Cookie.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>Cookie.SecondServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>s2</servlet-name>
    <url-pattern>/servlet2</url-pattern>
  </servlet-mapping>
     <servlet>
        <servlet-name>s3</servlet-name>
        <servlet-class>url.FirstServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>s4</servlet-name>
        <servlet-class>url.SecondServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>s3</servlet-name>
        <url-pattern>/servlet3</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>s4</servlet-name>
        <url-pattern>/servlet4</url-pattern>
    </servlet-mapping>
     <servlet>
        <servlet-name>s5</servlet-name>
        <servlet-class>hiddenform.FirstServlet</servlet-class>
    </servlet>

    <servlet>
        <servlet-name>s6</servlet-name>
        <servlet-class>hiddenform.SecondServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>s5</servlet-name>
        <url-pattern>/servlet5</url-pattern>
    </servlet-mapping>

    <servlet-mapping>
        <servlet-name>s6</servlet-name>
        <url-pattern>/servlet6</url-pattern>
    </servlet-mapping>
</web-app>
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Wt_HA</title>
</head>
<body>
<h1>Session Tracking</h1>
      <h2>Cookie Tracking</h2>
    <form action="servlet1" method="GET">
        Name:<input type="text" name="userName"/>
        <input type="submit" value="Submit" />
    </form>
    <h2>Url Rewriting</h2>
      <form action="servlet3">
        Name:<input type="text" name="userName" />
        <input type="submit" value="Submit" />
    </form>
    <h2>Hidden Form Field </h2>
     <form action="servlet5">
        Name:<input type="text" name="userName">
        <input type="submit" value="Submit" />
    </form>

</body>
</html>
```

Output:-



Session Tracking

Cookie Tracking

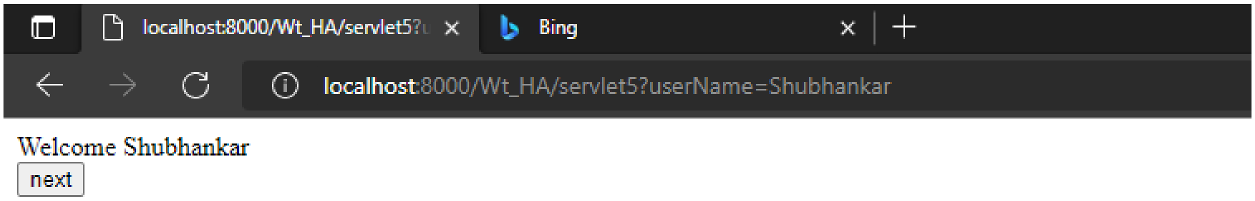
Name: Submit

Url Rewriting

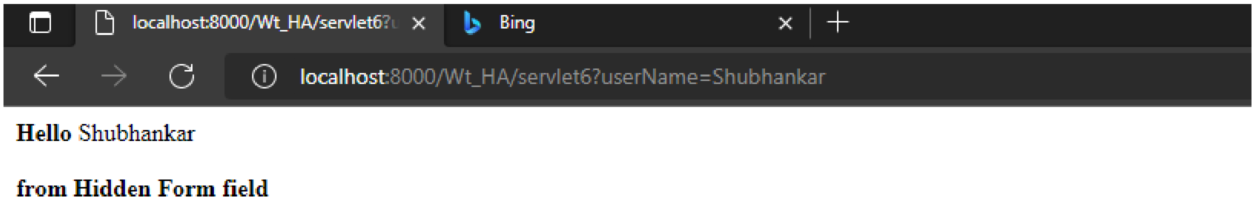
Name: Submit

Hidden Form Field

Name: Shubhankar Submit


localhost:8000/Wt_HA/servlet5?u
Bing
localhost:8000/Wt_HA/servlet5?userName=Shubhankar
next

Welcome Shubhankar


localhost:8000/Wt_HA/servlet6?u
Bing
localhost:8000/Wt_HA/servlet6?userName=Shubhankar

Hello Shubhankar

from Hidden Form field

Conclusion

In this design session tracking was done in java servlets using Cookie,URL rewriting and Hidden Filed methods.