



UT-02. Actividad 2.5 Sesiones

1. Objetivos

Actividad de descubrimiento conjunta para hacer control de sesiones. Basada en la documentación de <https://www.tutorialspoint.com/servlets>

2. Introducción

HTTP es un protocolo "sin estado", lo que significa que cada vez que un cliente recupera una página web, el cliente abre una conexión separada al servidor web y el servidor automáticamente no guarda ningún registro de la solicitud previa del cliente.

Todavía hay tres formas de mantener la sesión entre el cliente web y el servidor web:

2.1 Cookies

Un servidor web puede asignar una ID de sesión única como una cookie a cada cliente web y para las solicitudes posteriores del cliente pueden ser reconocidas utilizando la cookie recibida.

Puede que esta no sea una forma efectiva porque muchos navegadores temporales no admiten cookies, por lo que no recomendaría usar este procedimiento para mantener las sesiones.

2.2 Campos ocultos de forma

Un servidor web puede enviar un campo de formulario HTML oculto junto con una ID de sesión única de la siguiente manera:

```
<input type = "hidden" name = "sessionid" value = "12345">
```

Esta entrada significa que, cuando se envía el formulario, el nombre y el valor especificados se incluyen automáticamente en los datos GET o POST. Cada vez que el navegador web envía la solicitud de vuelta, se puede utilizar el valor de session_id para mantener la pista de diferentes navegadores web. Esta podría ser una forma efectiva de realizar un seguimiento de la sesión, pero al hacer clic en un enlace de hipertexto normal (<A HREF...>) no se envía un formulario, por lo que los campos ocultos del formulario tampoco pueden admitir el seguimiento general de la sesión.



2.3 Reescritura de URL

Puede agregar algunos datos adicionales al final de cada URL que identifica la sesión, y el servidor puede asociar ese identificador de sesión con los datos que ha almacenado sobre esa sesión. Por ejemplo, con `http://tutorialspoint.com/file.htm;sessionid = 12345`, el identificador de sesión se adjunta como `sessionid = 12345` al que se puede acceder en el servidor web para identificar al cliente. La reescritura de URL es una mejor manera de mantener las sesiones y funciona incluso cuando los navegadores no son compatibles con las cookies. El inconveniente de la reescritura de URL es que tendría que generar cada URL dinámicamente para asignar una ID de sesión, incluso en el caso de una página HTML estática simple.

2.4 El objeto HttpSession

Además de las tres formas mencionadas anteriormente, servlet proporciona HttpSession Interface, que proporciona una forma de identificar a un usuario en más de una solicitud de página o visitar un sitio web y almacenar información sobre ese usuario.

El contenedor de servlets usa esta interfaz para crear una sesión entre un cliente HTTP y un servidor HTTP. La sesión persiste durante un período de tiempo específico, a través de más de una solicitud de conexión o página del usuario. Obtendrá el objeto HttpSession llamando al método público `getSession ()` de `HttpServletRequest`, como se muestra a continuación:

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

Debe llamar a `request.getSession ()` antes de enviar cualquier contenido de documento al cliente. Aquí hay un resumen de los métodos importantes disponibles a través del objeto HttpSession:

Sr.No	Method & Description
1	public Object getAttribute(String name) This method returns the object bound with the specified name in this session, or null if no object is bound under the name.
2	public Enumeration getAttributeNames() This method returns an Enumeration of String objects containing the names of all the objects bound to this session.
3	public long getCreationTime() This method returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.



4	public String getId() This method returns a string containing the unique identifier assigned to this session.
5	public long getLastAccessedTime() This method returns the last accessed time of the session, in the format of milliseconds since midnight January 1, 1970 GMT
6	public int getMaxInactiveInterval() This method returns the maximum time interval (seconds), that the servlet container will keep the session open between client accesses.
7	public void invalidate() This method invalidates this session and unbinds any objects bound to it.
8	public boolean isNew() This method returns true if the client does not yet know about the session or if the client chooses not to join the session.
9	public void removeAttribute(String name) This method removes the object bound with the specified name from this session.
10	public void setAttribute(String name, Object value) This method binds an object to this session, using the name specified.
11	public void setMaxInactiveInterval(int interval) This method specifies the time, in seconds, between client requests before the servlet container will invalidate this session.



3. Ejemplo de seguimiento de sesión

Este ejemplo describe cómo usar el objeto `HttpSession` para conocer la hora de creación y la hora de último acceso para una sesión. Asociaremos una nueva sesión con la solicitud si aún no existe.

Crearemos nuestro java servlet llamado *SessionTrack*

```
// Import required java libraries
```

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

// Extend HttpServlet class
public class SessionTrack extends HttpServlet {

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

        // Create a session object if it is already not created.
        HttpSession session = request.getSession(true);

        // Get session creation time.
        Date createTime = new Date(session.getCreationTime());

        // Get last access time of this web page.
        Date lastAccessTime = new Date(session.getLastAccessedTime());

        String title = "Welcome Back to my website";
        Integer visitCount = new Integer(0);
        String visitCountKey = new String("visitCount");
        String userIDKey = new String("userID");
        String userID = new String("ABCD");

        // Check if this is new comer on your web page.
        if (session.isNew()) {
            title = "Welcome to my website";
```



```
session.setAttribute(userIDKey, userID);
} else {
    visitCount = (Integer)session.getAttribute(visitCountKey);
    visitCount = visitCount + 1;
    userID = (String)session.getAttribute(userIDKey);
}
session.setAttribute(visitCountKey, visitCount);

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

String docType =
    "<!doctype html public \"-//w3c//dtd html 4.0 \" +
    \"transitional//en\">\n";

out.println(docType +
    "<html>\n" +
    "<head><title>" + title + "</title></head>\n" +

    "<body bgcolor = \"#f0f0f0\">\n" +
    "<h1 align = \"center\">" + title + "</h1>\n" +
    "<h2 align = \"center\">Session Infomation</h2>\n" +
    "<table border = \"1\" align = \"center\">\n" +

    "<tr bgcolor = \"#949494\">\n" +
    "    <th>Session info</th><th>value</th>
    </tr>\n" +

    "<tr>\n" +
    "    <td>id</td>\n" +
    "    <td>" + session.getId() + "</td>
    </tr>\n" +

    "<tr>\n" +
    "    <td>Creation Time</td>\n" +
    "    <td>" + createTime + "    </td>
```



```
        </tr>\n" +\n\n        "<tr>\n" +\n        "    <td>Time of Last Access</td>\n" +\n        "    <td>" + lastAccessTime + "    </td>\n" +\n        </tr>\n" +\n\n        "<tr>\n" +\n        "    <td>User ID</td>\n" +\n        "    <td>" + userID + "    </td>\n" +\n        </tr>\n" +\n\n        "<tr>\n" +\n        "    <td>Number of visits</td>\n" +\n        "    <td>" + visitCount + "</td>\n" +\n        </tr>\n" +\n        "</table>\n" +\n        "</body>\n" +\n        "</html>"\n    );\n}\n}
```

Compile el Servidor SessionTrack anterior y cree una entrada apropiada en el archivo web.xml. Ahora ejecutar `http: // localhost: 8080 / SessionTrack` mostraría el siguiente resultado cuando se ejecutaría por primera vez:



Welcome to my website

Session Infomation

Session info	value
id	0AE3EC93FF44E3C525B4351B77ABB2D5
Creation Time	Tue Jun 08 17:26:40 GMT+04:00 2010
Time of Last Access	Tue Jun 08 17:26:40 GMT+04:00 2010
User ID	ABCD
Number of visits	0

Y lo volvemos a correr pasados unos segundos.

Welcome Back to my website

Session Infomation

info type	value
id	0AE3EC93FF44E3C525B4351B77ABB2D5
Creation Time	Tue Jun 08 17:26:40 GMT+04:00 2010
Time of Last Access	Tue Jun 08 17:26:40 GMT+04:00 2010
User ID	ABCD
Number of visits	1



4. Eliminar datos de sesión.

Cuando haya terminado con los datos de sesión de un usuario, tiene varias opciones:

Eliminar un atributo en particular: puede llamar al método `public void removeAttribute (String name)` para eliminar el valor asociado con una clave en particular.

Eliminar toda la sesión: puede llamar al método `public void invalidate ()` para descartar una sesión completa.

Configuración del tiempo de espera de la sesión: puede llamar al método `public void setMaxInactiveInterval (int interval)` para establecer el tiempo de espera de una sesión individualmente.

Iniciar sesión en el usuario: los servidores que admiten Servlets 2.4, puede llamar al cierre de sesión para desconectar al cliente del servidor web e invalidar todas las sesiones que pertenecen a todos los usuarios.

Configuración web.xml: si usa Tomcat, aparte de los métodos mencionados anteriormente, puede configurar el tiempo de espera de la sesión en el archivo web.xml de la siguiente manera:

```
<session-config>
  <session-timeout>15</session-timeout>
</session-config>
```

The timeout is expressed as minutes, and overrides the default timeout which is 30 minutes in Tomcat.

The `getMaxInactiveInterval()` method in a servlet returns the timeout period for that session in seconds. So if your session is configured in web.xml for 15 minutes, `getMaxInactiveInterval()` returns 900.



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package es.cifpcm.sesion1;

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

/**
 *
 * @author inmav
 */
public class SesionTrackServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */

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

        // Crea un objeto sesión si este no existe
    }
}
```



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

//Obtiene el tiempo de inicio de sesión
Date createTime = new Date(session.getCreationTime());

// Obtiene el momento de acceso de la última vez
Date lastAccessTime = new Date(session.getLastAccessedTime());

String title = "Bienvenido de nuevo a mi página";
Integer visitCount = 0;
String visitCountKey = "visitCount";
String userIDKey = "userID";
String userID = "ABCD";

// Comprueba si eres nuevo
if (session.isNew()) {
    title = "Bienvenido a mi página WEB";
    session.setAttribute(userIDKey, userID);
} else {
    visitCount = (Integer)session.getAttribute(visitCountKey);
    visitCount = visitCount + 1;
    userID = (String)session.getAttribute(userIDKey);
}
session.setAttribute(visitCountKey, visitCount);

// Obtiene response generando la salida en forma de código html
response.setContentType("text/html");
PrintWriter out = response.getWriter();

String docType =
    "<!doctype html public \"-//w3c//dtd html 4.0 \" +
    \"transitional//en\">\n";

out.println(docType +
```



```
"<html>\n" +
  "<head><title>" + title + "</title></head>\n" +

  "<body bgcolor = \"#f0f0f0\">\n" +
    "<h1 align = \"center\">" + title + "</h1>\n" +
    "<h2 align = \"center\">Información sobre la sesión</h2>\n" +
    "<table border = \"1\" align = \"center\">\n" +

      "<tr bgcolor = \"#949494\">\n" +
        "  <th>Información de sesión</th><th>value</th>"+
      "</tr>\n" +

      "<tr>\n" +
        "  <td>id</td>\n" +
        "  <td>" + session.getId() + "</td>"+
      "</tr>\n" +

      "<tr>\n" +
        "  <td>Inicio Sesión</td>\n" +
        "  <td>" + createTime + " </td>"+
      "</tr>\n" +

      "<tr>\n" +
        "  <td>Último acceso</td>\n" +
        "  <td>" + lastAccessTime + " </td>"+
      "</tr>\n" +

      "<tr>\n" +
        "  <td>User ID</td>\n" +
        "  <td>" + userID + " </td>"+
      "</tr>\n" +
      "<tr>\n" +
        "  <td>Número de visitas</td>\n" +
        "  <td>" + visitCount + "</td>"+
```



```
        "</tr>\n" +  
        "</table>\n" +  
        "</body>" +  
        "</html>"  
    );  
}  
}
```