**HttpSessionListener**

Antes de entrar en los detalles de la **SessionListener** debemos saber en primer lugar de las sesiones. Como sabemos que el protocolo HTTP es un protocolo *"sin estado".* El término sin estado significa que no puede persistir la información. No puede recordar las transacciones anteriores. Cada vez que un cliente realiza una solicitud para cualquier recurso a la servidor, el servidor recibe la solicitud y procesa la solicitud y envía de vuelta la respuesta. Después de enviar la respuesta del servidor cierra la conexión y se olvida de las solicitudes anteriores. Cada vez que un cliente envía ninguna solicitud al servidor, el servidor trata cada petición como una nueva solicitud. Para eliminar esta nos ha facilitado la instalación de la sesión. En el seguimiento de sesión cada vez que un cliente envía una solicitud al servidor, a continuación servidor crea un identificador único para esa solicitud y la envía de vuelta la ID única para el cliente junto con el objeto de respuesta, ahora cada vez que un cliente envía una solicitud al servidor también envía una Identificación única con él para que el servidor pueda saber de dónde proviene la solicitud.

**Los oyentes** escuchan los eventos. Cada vez que se produce cualquier evento que se escucha por el oyente. El oyente será controlador de los servidores web.

***HttpSessionListener*** es una interfaz que extiende la clase **java.util.EventListener.** El propósito principal de este oyente es notificar a cada vez que hay un cambio en la lista de sesiones activas en una aplicación web

Esta interfaz tiene dos métodos:

1. **sessionCreated (evento HttpSessionEvent):** Se le notificará cuando se crea la sesión.
2. **sessionDestroyed (evento HttpSessionEvent):** Se le notificará cuando la sesión queda invalidada.

En los métodos anteriores, podemos ver que hemos usado **HttpSessionEvent** como parámetro de los dos métodos. Esta clase tiene un solo método **getSession ()** que devuelve la sesión actual.

**El código del programa es la siguiente:**

Hacer la entrada de este archivo en el archivo descriptor de despliegue web.xml que es

|  |
| --- |
| importar javax.servlet \*.;  importar javax.servlet.http \*.;   ListenerSession clase pública  implementa HttpSessionListener {  ListenerSession pública () {  }  public void sessionCreated (sessionEvent HttpSessionEvent) {  / / Obtener el período de sesiones que se creó  HttpSession session = sessionEvent.getSession ();  / / Algo en la sesión de la tienda, y registrar un mensaje  try {  System.out.println ("Sesión de creación:" + período de sesiones);  session.setAttribute ("perro", "labrador");  session.setAttribute ("nombre", "Diana");  } Catch (Exception e) {  System.out.println ("Error al establecer atributo de sesión:" +  e.getMessage ());  }  }  public void sessionDestroyed (sessionEvent HttpSessionEvent) {  / / Obtener la sesión que fue invalidado  HttpSession session = sessionEvent.getSession ();  / / Registrar un mensaje  System.out.println ("Sesión invalidado:" + período de sesiones);  System.out.println ("La raza del perro es:" + session.getAttribute ("perro"));  System.out.println ("El nombre del perro es:" + session.getAttribute ("nombre"));  }  } |

|  |
| --- |
| importar javax.servlet \*.;  importar javax.servlet.http \*.;  import java.io. \*;   ServletListenerSession clase pública se extiende HttpServlet {  public void doGet (HttpServletRequest request,  Respuesta HttpServletResponse) throws ServletException, IOException  {  response.setContentType ("text / html");  PrintWriter pw = response.getWriter ();  HttpSession session = request.getSession ();  String str = (String) session.getAttribute ("perro");  Cadena dogName = (String) session.getAttribute ("nombre");  pw.println ("La raza del perro es" + str);  pw.println ("El nombre del perro es" + dogName);  }  } |