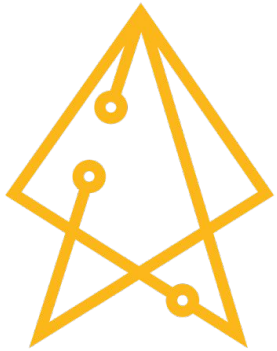
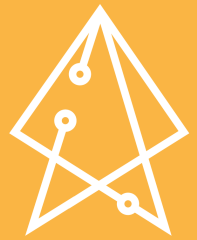


ACADEMIK

Java EE™

Sesiones



ACADEMIK

¿Cómo identificar las conexiones de cada clientes?

Existen diferentes “técnicas” para tener control de las conexiones

Sesiones en HTTP

El protocolo HTTP es un protocolo de comunicación **“stateless”**, esto significa que no almacena estado, es decir que no se tiene un historial de las peticiones realizadas por cada cliente.

Entonces, el seguimiento de las conexiones de cada usuario se debe realizar con estrategias independientes del protocolo HTTP, entre las técnicas más usadas están:

- HTML hidden field
- Uso de cookies
- API oficial de Java HttpSession
- URL rewriting

HTML Hidden Field

Esta técnica consiste en crear un campo oculto único en el HTML y cuando el usuario comienza a navegar, se establece un valor único para cada usuario y así poder realizar un seguimiento de la sesión.

Problema: Este método no se puede usar con enlaces porque necesita que se envíe el formulario cada vez que se realiza una solicitud del cliente al servidor con el campo oculto. Tampoco es seguro porque podemos obtener el valor del campo oculto de la fuente HTML y usarlo para hackear la sesión.

Cookies

Las cookies son pequeñas piezas de información que el servidor web envía en el encabezado de las respuestas y se almacena en las cookies del navegador. Cuando el cliente realiza una petición adicional, agrega la cookie al encabezado de dicha petición y así se puede realizar un seguimiento de la sesión.

Problema: Si el cliente desactiva las cookies, el seguimiento de sesión no funcionará.

URL rewriting

Esta técnica consiste en agregar un parámetro de identificador de sesión con cada petición y respuesta para realizar un seguimiento de la sesión.

Problema: Esto es muy tedioso porque debemos mantener un registro de este parámetro en cada respuesta y asegurarnos de que no coincida con otros parámetros, de igual forma es posible que el usuario modifique el valor y haga una especie de hack tratando de utilizar el ID de otro usuario.

Java HttpSession API

La API de administración de sesión de Java se basa en los métodos anteriores para el seguimiento de sesión.

Además del seguimiento de sesiones, esta API permite almacenar algunos datos adicionales en la sesión que podemos usar en futuras solicitudes.

Problema: Para garantizar el funcionamiento del seguimiento de sesiones, aún con las cookies deshabilitadas, se debe combinar con URL rewriting

Login utilizando Cookies

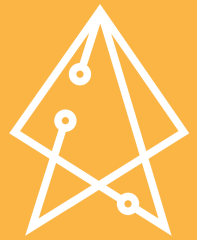
```
@WebServlet("/login-cookies")
public class LoginWithCookiesServlet extends HttpServlet {

    @Override
    public void doPost(
        HttpServletRequest req, HttpServletResponse resp
    ) throws ServletException, IOException {
        // Do login and create a cookie
        Cookie cookie = new Cookie("user",user);
        cookie.setMaxAge(10*60);
        response.addCookie(cookie);
        // And then redirect the response
    }
}
```

Login utilizando HttpSession

```
@WebServlet("/login-session-api")
public class LoginServlet extends HttpServlet {

    @Override
    public void doPost(
        HttpServletRequest req, HttpServletResponse resp
    ) throws ServletException, IOException {
        // Do login and create a session
        HttpSession session = request.getSession();
        session.setAttribute("user", "Signed user");
        session.setMaxInactiveInterval(30*60);
        // And then redirect the response
    }
}
```



ACADEMIK

¿Cómo funciona HttpSession API?

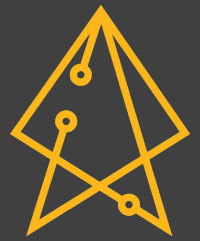
Respuesta corta: Una cookie muy particular

La interfaz **HttpSession**

El API de Servlets de Java provee una interfaz para la administración de sesiones, HttpSession, disponible en el objeto HttpServletRequest por medio de los métodos getSession() y getSession(boolean create), esta interfaz permite “guardar” objetos relacionados a cada sesión que pueden ser obtenidos nuevamente para utilizarlo en próximas peticiones.

getSession(): Retorna un objeto de tipo HttpSession. Si la petición no tiene vinculada una sesión, crea una nueva.

getSession(boolean create): Retorna un objeto de tipo HttpSession. Si la petición no tiene vinculada una sesión y la bandera create es verdadera, crea una nueva sesión. De lo contrario, si la petición no tiene vinculada una sesión y la bandera es falsa, retorna null.



ACADEMIK

Gracias por su atención

Escríbenos a: cursos@nabenik.com

www.academik.io