

ALUMNO: MARIA GONZALEZ HERRERO

Asignatura: Programación de Sistemas Distribuidos

Curso: 2021/2022
Semestre: 2º

Fecha: 08-02-2022

Ejercicio práctico clase:

Tomando el fichero que tenemos en el campus sockets_alumno.html

1. Ábrelo con el navegador en tú ordenador. ¿Qué aparece en pantalla? ¿Y en consola?

En la pantalla aparece así:

WebSocket Test

En la consola aparece así:

```
✖ ▶ WebSocket connection to 'ws://nebrija.sockets.com/v2/test' failed: sockets_alumno.html:25
```

No se conecta al socket porque hay un problema, es decir, la dirección no existe

2. ¿Qué deberías hacer cambiar el output de la aplicación en el estado anterior?

Cambiar la dirección que había, la cual no existía por una nueva pública.

```
var wsUri = "wss://demo.piesocket.com/v3/channel_1?api_key=oCdCMcMPQpbvNjUIzqtVF1d2X2okWpDQj4AwARJuAgtjhzKxVEjQU6IdCjwm&notify_self"
```

En el html sale así:

WebSocket Test

CONNECTED

SENT: Hello world

RESPONSE: {"info":"You are using a test api key"}

RESPONSE: Hello world

RESPONSE: Hello world

RESPONSE: Hello world

RESPONSE: Hello world

RESPONSE: Hello world

RESPONSE: a estudiar

RESPONSE: Hello world

RESPONSE: Hello world



3. ¿Para qué sirve `window.addEventListener`?

Es un escuchador que indica al navegador que esté atento a la interacción del usuario.

En el código de nuestro ejercicio aparece al final de este de esta forma:

```
window.addEventListener("load", init, false); // se ejecuta cuando se carga la pagina
```

4. Tomando como referencia lo que tenemos en la función `onMessage` ¿Cómo completarías la función `onError(evt)`? ¿Debes realizar la llamada a esta función en otra parte de la aplicación?

De esta forma completaría la función `onError(evt)`:

```
function onClose(evt) // funcion que se ejecuta cuando se cierra el socket
{
    writeLog("DISCONNECTED");
    websocket.close();
}
```

Teniendo en cuenta la función de `onOpen()`, que se llama en la función de `testWebSockets()`, para ver que se abre el socket, haremos lo mismo pero para comprobar de que hemos cerrado la conexión, quedando el código de esta forma:

```
function testWebSocket()
{
    websocket = new WebSocket(wsUri);
    websocket.onopen = function(evt) { onOpen(evt) }; //cuando se abre el socket
    websocket.onmessage = function(evt) { onMessage(evt) };
    websocket.onclose = function(evt) { onClose(evt) }; //esto es para cuando se cierra la conexion
    websocket.onmessage = function(evt) { onMessage(evt) };
    websocket.onError = function(evt) { onError(evt) };
}
```

5. ¿Cómo ultimarías la función onClose(evt)?

```
function onClose(evt) // funcion que se ejecuta cuando se cierra el socket
{
    | writeLog("DISCONNECTED");
}
```

Teniendo en cuenta la función ya escrita de onOpen(), la función onClose() se ejecutará cuando se cierra el socket.

6. Investiga en internet sobre XML y busca su sintaxis y relación con los sistemas distribuidos actuales.

XML se dio a conocer por primera vez entorno a los años 70 por IBM, se creó debido a la necesidad que tenían las empresas para guardar grandes cantidades de información y poder compartirla en otros Sistemas Operativos y plataformas. XML (Extensible Markup Language), es un metalenguaje que se encarga de definir unas marcas legibles para otros lenguajes, proviene de su antecesor el lenguaje SGML y permite definir la gramática de los lenguajes específicos para poder estructurar documentos grandes.

Al igual que otros lenguajes XML tiene ventajas y desventajas frente a otros.

Ventajas:

- Es extensible, es decir, una vez diseñado y puesto en producción, se puede extender su utilización mediante etiquetas, haciendo que su uso se simplifique más
- Su analizador es estándar, porque no es necesario crear un analizador específico para cada versión del lenguaje, es muy útil para poder evitar “bugs” y acelerar el desarrollo de las aplicaciones.
- Si alguien externo al proyecto quiere usar el documento, su estructura sencilla hace más fácil el procesado, se puede comunicar entre diferentes plataformas o Sistemas Operativos.

Desventajas:

- Las bases de datos de XML son más lentas y por ello necesitan que los datos estén comprimidos para así funcionar de una forma más rápida y precisa.
- Las búsquedas son mas lentas que en una base de datos relacional, ya que se organizan a través de texto y de etiquetas.
- Hay cierta limitación en cuanto se refiere a los gestores de las bases de datos que pueden utilizar XML



- Los documentos de XML no están preparados para almacenar información a largo plazo ya que con el tiempo la información guardada en ellos se deteriora.
- La seguridad de los datos no está garantizada, por ejemplo, no puedes saber quien esta actualizado la base de datos o realizando algún cambio en ella.

Desde su inicio a tenido grandes iniciativas lo que la han hecho ponerse al nivel de aceptación que otros lenguajes, además de haber tenido referencia tanto en ámbitos académicos, empresariales o institucionales.

Durante el 2004 se puede citar una continua publicación de nuevas versiones informáticas centradas en documentos con XML, diseñando nuevos vocabularios o la adopción del lenguaje en distintos ámbitos de trabajo.

Un mayor respaldo por parte de los organismos de normalización internacional a las especificaciones distintas a lograr la estandarización del comercio electrónico entre empresas. Y la prueba de ello está en las nuevas actualizaciones del estándar ISO y del estándar UDDI.

Gracias a estas nuevas actualizaciones se ha vuelto a despertar el interés hacia XML.