UNIVERSIDAD NEBRIJA

ALUMNO:

Asignatura: Programación de Sistemas Distribuidos

Curso: 2021/2022 Fecha: 08-02-2022
Semestre: 2º

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?**

   Al ejecutar el programa, nos da un error en la consola, el cuál nos indica que no se conecta al socket porque la dirección url no existe, para solucionar dicho error, buscamos una dirección de un socket público y la cambiamos.

   Y en pantalla aparece el título websocket test.

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

   Buscar una dirección web socket pública y reemplazarla por la inicial que tenemos en la variable wsUri.

```
13
14      var wsUri = "wss://demo.piesocket.com/v3/channel_1?api_key=oCdCMcMPQpbvNjUIzqtvF1d2X2okWpDQj4AwARJuAgtjhzKxVEjQU6IdCjwm&notify_self"
```

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

   Esta función espera a un evento del tipo “load”, recibe una notificación cuando dicho evento ocurre y llama una función java de tipo init.

```
65      window.addEventListener("load", init, false);
```

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?**

   Para completar la función la función onError(evt), añadimos un writelog y además realizar la llamada dicha función en la función testWebSocket().

```
function onError(evt)
{
  writeLog('<span style="color: red;">Error! </span>');
}
```

```
function testWebSocket()
{
  websocket = new WebSocket(wsUri);
  websocket.onopen    = function(evt) { onOpen(evt)    };
  websocket.onmessage = function(evt) { onMessage(evt) };
  websocket.onerror   = function(evt) { onError(evt) };
}
```

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

Para terminar la función onClose, añadimos un writelog que ponga disconnet y enviamos un mensaje de despedida.
Además de añadir la función a la función testeWebSocket() al igual que las anteriores.

```
function onClose(evt)
{
  writeLog("DISCONNECT");
  sendMessage("bye world");
}
```

```
function testWebSocket()
{
  websocket = new WebSocket(wsUri);
  websocket.onopen    = function(evt) { onOpen(evt)    };
  websocket.onmessage = function(evt) { onMessage(evt) };
  websocket.onerror   = function(evt) { onError(evt) };
  websocket.onclose = function(evt) {onClose(evt)};
}
```

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

   XML, son las siglas de Extensible Markup Language, que se puede traducir como lenguaje de marcas extensible. Los archivos XML se componen de etiquetas que aportan datos e información a procesar, estas etiquetas pueden estar de forma individual o anidada. Esto quiere decir que dentro de una etiqueta podemos tener a su vez una o mas etiquetas. Además cada etiqueta debe de estar correctamente cerrada, nunca deben acotarse, cada una tiene su inicio y su final, si eliminamos alguna de estas partes nos daría error.

   La relación de este lenguaje con los sistemas distribuido es que en los sistemas distribuidos se pueden compartir recursos de un dispositivo a otro, por lo que este lenguaje es de gran utilidad a la hora de compartir código, ya que como hemos comentado se basa en etiquetas, que dan información sobre el código que esta escrito, lo que facilitara en gran medida la compartición de dicho código o información.