

WebSockets

Desarrollo de Aplicaciones Web

Juan Francisco Moreno Alguacil

David Galisteo Cantero

Contenido

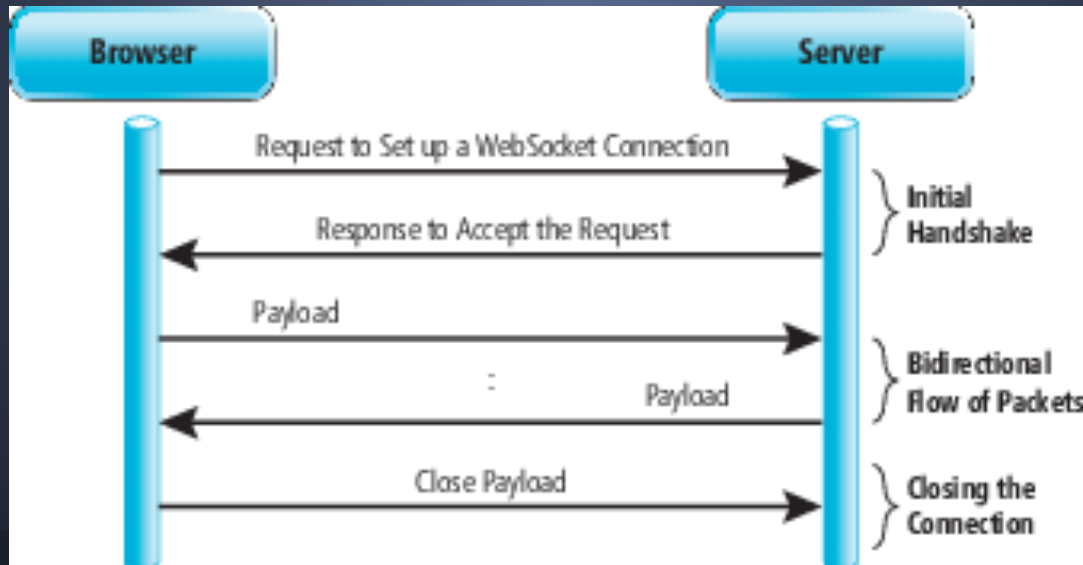
- ¿Qué es 'WebSocket'?
- ¿Cómo funciona?
- Ajax y WebSocket
- Código ejemplo
- Video demostración
- Conclusiones
- Fuentes

¿Qué es WebSocket?

- WebSocket es un protocolo que define una comunicación bidireccional sobre un mismo socket TCP, a través del cual, tanto cliente como servidor, pueden enviar mensajes.
- La conexión se mantiene durante todo el tiempo, puede parecer que consume gran cantidad de recursos, pero no es así.
- Lo fantástico de esta tecnología es que los datos y notificaciones pueden ir y venir entre cliente y servidor, sin retrasos ni solicitudes adicionales.
- En Diciembre de 2009 Chrome se convertía en el primer navegador en ofrecer soporte para WebSockets (No era un estándar)

¿Cómo funciona?

- La comunicación comienza con un 'handshake' en el que se especifica que ambas partes usarán este protocolo. Las peticiones son HTTP, en ellas se pide un 'upgrade' para usar el protocolo WS.



Ajax y WebSocket

- Se ha hablado de que vendría a sustituir a la tecnología Ajax.
- Una de las ventajas de WebSocket respecto a Ajax es la comunicación bidireccional.
- Sin embargo, dependiendo de la aplicación que queramos construir, usaremos uno u otro. (Live Apps or 'Hot' Data)



Azul: Ajax
Rojo: WebSocket

Código ejemplo (Muro real time)

El muro

Mensaje:

Enviar

Conectado como David

David: hola

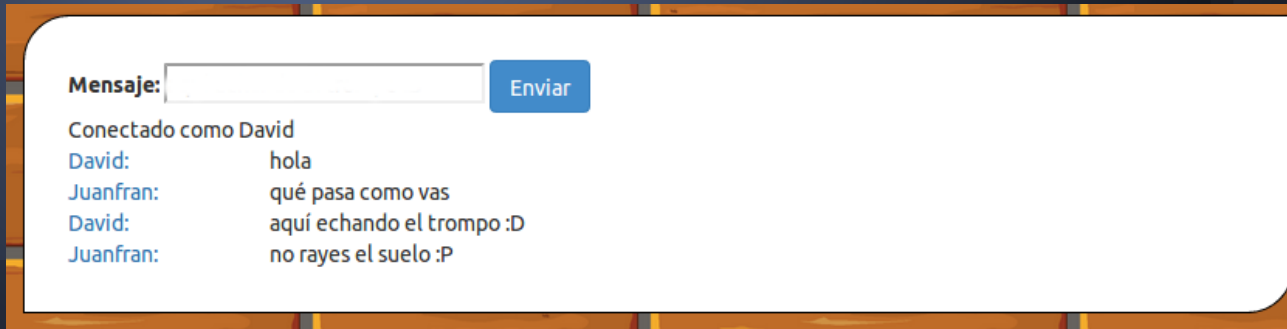
Juanfran: qué pasa como vas

David: aquí echando el trompo :D

Juanfran: no rayes el suelo :P

Código ejemplo (HTML Cliente)

```
<h1>El muro</h1>
<div class="wrapper">
  <form>
    <label for="mensaje">Mensaje:</label>
    <input id="mensaje" name="mensaje" />
    <button class="btn btn-primary" type="button" onclick="
enviarMensaje()">Enviar</button>
  </form>
</div>
<div id="muro">
</div>
```



The screenshot shows a web form and a chat interface. The form has a label "Mensaje:" followed by a text input field and a blue "Enviar" button. Below the form, the chat interface shows a status "Conectado como David" and a list of messages:

- David: hola
- Juanfran: qué pasa como vas
- David: aquí echando el trompo :D
- Juanfran: no rayes el suelo :P

Código ejemplo (JS cliente)

Ejemplo, evitar este código en producción (XSS)



```
var nick = window.prompt("Cual es tu nombre?", "Default");
var wsUri = "ws://localhost:8080/WebSocket-Galisteo-Moreno/muro";
var websocket = new WebSocket(wsUri);
//eventos
websocket.onopen = function(evt){
    log("<div class='msg'>Conectado como "+nick+"</div>");
}
websocket.onmessage = function(evt){
    log(evt.data);
}
websocket.onerror = function(evt){
    log("<div class='error'> Se ha producido un error: </div>" + evt.data);
}
```

```
function enviarMensaje(){
    websocket.send("<div class='user'>"+nick+"</div> <div class='msg'>"+message.value+"</div>");
}

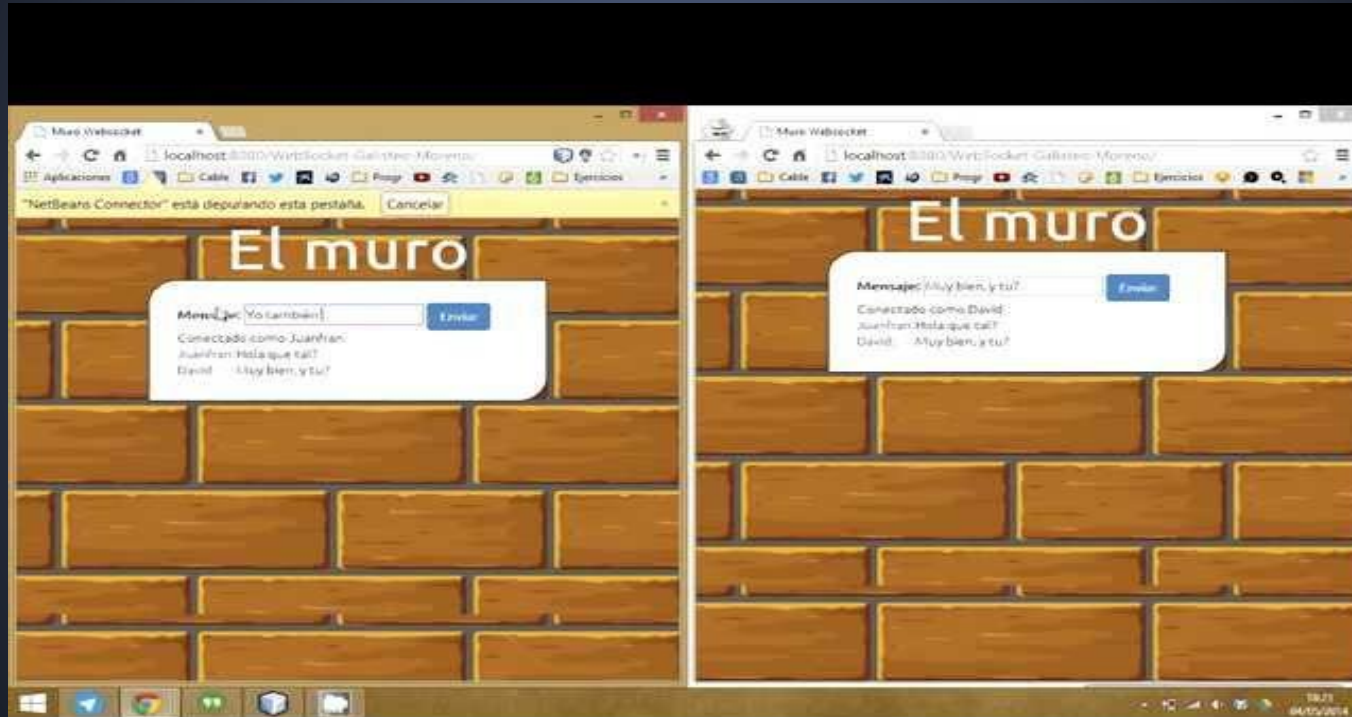
function log(mensaje){
    var logDiv = document.getElementById("muro");
    logDiv.innerHTML += ("<div class='mensaje'>"+mensaje+"</div>");
}
```


Código ejemplo (Java Servidor)

```
@ServerEndpoint("/muro")
@Singleton
public class ControladorMuro { ... }
//Lista de conexiones
static final List<Session> conexiones = new ArrayList<>();
//Cuando un cliente se conecta
@OnOpen
public void iniciaSesion(Session session) {
    conexiones.add(session);
}
//Se recibe un mensaje y se notifica a todos los usuarios
@OnMessage
public void onMessage(String mensaje, Session sesion) { notificar(mensaje); }
```

```
//Cliente se desconecta
@OnClose
public void finConexion(Session session){
    if (conexiones.contains(session)) {
        session.close();
        conexiones.remove(session);
    }
}
//notificación broadcast
public void notificar(String mensaje) {
    for (Session sesion : conexiones) {
        RemoteEndpoint.Basic remote = sesion.
        getBasicRemote();
        remote.sendText(mensaje);
    }
}
```

Video demostración



<https://www.youtube.com/watch?v=N7o5ja-pBjk>

Conclusiones

- Se ha demostrado que el protocolo WS es vulnerable (Cross Site Request Forgery), por lo que algunos navegadores lo han deshabilitado, y aún hay que esperar a que esta tecnología termine de ‘madurar’.
 - https://www.owasp.org/index.php/Testing_WebSockets
- Si piensas desarrollar una aplicación con esta tecnología, sopesa las ventajas e inconvenientes, una mala decisión podría acarreararte problemas :D

Fuentes

- <http://www.websocket.org/aboutwebsocket.html>
- <http://msdn.microsoft.com/es-es/magazine/hh975342.aspx>
- <http://www.desarrolloweb.com/articulos/websockets.html>
- <http://www.apuntesdejava.com/2013/05/websockets-en-java-ee-7-jsr-356.html>