



EBook Gratis

APRENDIZAJE WebSockets

Free unaffiliated eBook created from
Stack Overflow contributors.

#websocket

Tabla de contenido

Acerca de	1
Capítulo 1: Empezando con WebSockets	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Ejemplo de cliente web-socket.....	2
Capítulo 2: Diferencias entre WebSockets y AJAX	4
Introducción.....	4
Examples.....	4
ejemplo recibiendo notificación.....	4
Creditos	6

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [websockets](#)

It is an unofficial and free WebSockets ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official WebSockets.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con WebSockets

Observaciones

Esta sección proporciona una descripción general de qué es websocket y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema importante dentro de websocket y vincular a los temas relacionados. Dado que la Documentación para websocket es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar un socket web.

Ejemplo de cliente web-socket

Este es un ejemplo de un cliente web-socket en javascript.

Eso:

1. Se conecta a un servidor de demostración en vivo.
2. Envía un mensaje.
3. Recibe mensaje (s).
4. Se desconecta después de un intervalo.

```
var mySocket      = null;
var serverUrl     = 'wss://echo.websocket.org'; // wss: is ws: but using SSL.
var oWebSocket    = window.WebSocket || window.MozWebSocket;
if (oWebSocket) {
  mySocket = new oWebSocket (serverUrl);
  if (mySocket) {
    console.log (mySocket);
    mySocket.onopen      = onSocketOpen;
    mySocket.onclose     = onSocketClose;
    mySocket.onmessage   = onSocketMessage;
    mySocket.onerror     = onSocketError;

    setTimeout (closeSocket, 5000); // Be polite and free socket when done.
  }
}

function onSocketOpen (evt) {
  console.log ("Socket is now open.");
  mySocket.send ("Hello from my first live web socket!");
}

function onSocketClose (evt) {
  console.log ("Socket is now closed.");
}
```

```
}

function onSocketMessage (evt) {
    console.log ("Recieved from socket: ", evt.data);
}

function onSocketError (evt) {
    console.log ("Error with/from socket!");
    console.log (evt);
}

function closeSocket () {
    if (mySocket.readyState !== mySocket.CLOSED) {
        console.log ("Closing socket from our end (timer).");
        mySocket.close ();
    }
    else
        console.log ("Socket was already closed (timer).");
}
}
```

Si ejecuta ese código en una consola, obtiene un resultado como:

```
WebSocket { url: "wss://echo.websocket.org/", readyState: 0, bufferedAmount: 0, onopen: null,
  onerror: null, onclose: null, extensions: "", protocol: "", onmessage: null,
  binaryType: "blob"
}
2 // <--- 2 was the socket ID in this instance.
Socket is now open.
Recieved from socket: Hello from my first live web socket!
Closing socket from our end (timer).
Socket is now closed.
```

Lea Empezando con WebSockets en línea:

<https://riptutorial.com/es/websocket/topic/7958/empezando-con-websockets>

Capítulo 2: Diferencias entre WebSockets y AJAX

Introducción

Ajax usa el protocolo HTTP y puede enviar solicitudes usando métodos POST / GET de cliente a servidor.

WebSocket es en sí mismo un protocolo para comunicarse entre el Cliente y el Servidor, distinto de HTTP.

En Ajax, cuando envía una solicitud, el servidor envía una respuesta para esa solicitud y la conexión finaliza.

Al usar WebSockets cuando establece una conexión con el servidor, puede comunicarse entre el cliente y el servidor tanto como desee y mantiene la conexión activa.

Examples

ejemplo recibiendo notificación

Ajax:

1. Nueva solicitud :

- Cliente envía solicitud: Servidor, ¿Tiene nueva notificación?
- Servidor envía respuesta: Cliente, NO
- termina la conexión

2. Nueva solicitud :

- Cliente envía solicitud: Servidor, ¿Tiene nueva notificación?
- Servidor envía respuesta: Cliente, NO
- termina la conexión

3. Nueva solicitud :

- Cliente envía solicitud: Servidor, ¿Tiene nueva notificación?
- Servidor envía respuesta: Cliente, NO
- termina la conexión

4. Nueva solicitud :

- Cliente envía solicitud: Servidor, ¿Tiene nueva notificación?
- Servidor envía respuesta: Cliente, sí, aquí estás.
- termina la conexión

¡Ves que hay una cantidad de peticiones inútiles!

web-socket

- conexión establecida :

- cliente dice: Servidor, ¿Tiene nueva notificación?
después de xx pasó el tiempo
- El servidor dice: si tengo
después de xx pasó el tiempo
- El servidor dice: si tengo
- El cliente dice: Servidor, envía un nuevo correo electrónico si tengo
después de xx pasó el tiempo
- El servidor dice: ahora tienes nuevo correo electrónico

Ahora está claro que con web-socket no tendremos solicitudes inútiles

Lea Diferencias entre WebSockets y AJAX en línea:

<https://riptutorial.com/es/websocket/topic/8139/diferencias-entre-websockets-y-ajax>

Creditos

S. No	Capítulos	Contributors
1	Empezando con WebSockets	Brock Adams , Community , Farhad
2	Diferencias entre WebSockets y AJAX	Farhad , jalanb