



EBook Gratuito

APPENDIMENTO

WebSockets

Free unaffiliated eBook created from
Stack Overflow contributors.

#websocket

Sommario

Di.....	1
Capitolo 1: Iniziare con WebSockets	2
Osservazioni.....	2
Examples.....	2
Installazione o configurazione.....	2
esempio di client web-socket.....	2
Capitolo 2: Differenze tra WebSockets e AJAX	4
introduzione.....	4
Examples.....	4
esempio ricevendo la notifica.....	4
Titoli di coda	6

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

Capitolo 1: Iniziare con WebSockets

Osservazioni

Questa sezione fornisce una panoramica di cosa sia websocket e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di websocket e collegarsi agli argomenti correlati. Poiché la Documentation for websocket è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Installazione o configurazione

Istruzioni dettagliate su come installare o installare la presa di rete.

esempio di client web-socket

Questo è un esempio di un client web-socket in javascript.

It:

1. Si collega a un server demo dal vivo.
2. Invia un messaggio.
3. Riceve messaggi (s).
4. Si disconnette dopo un intervallo.

```
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).");
}
}
```

Se si esegue quel codice in una console si ottiene un risultato come:

```
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.
```

Leggi Iniziare con WebSockets online: <https://riptutorial.com/it/websocket/topic/7958/iniziare-con-websockets>

Capitolo 2: Differenze tra WebSockets e AJAX

introduzione

Ajax utilizza il protocollo HTTP e può inviare richieste utilizzando i metodi POST / GET da Client a Server.

WebSocket è esso stesso un protocollo per comunicare tra Client e Server, distinto da HTTP.

In Ajax quando si invia una richiesta, il server invia la risposta per tale richiesta e la connessione termina.

Usando WebSockets quando stabilisci una connessione con il server, puoi comunicare tra client e server quanto vuoi e mantiene attiva la connessione.

Examples

esempio ricevendo la notifica

Ajax:

1. Nuova richiesta :

- Il cliente invia una richiesta: Server, hai una nuova notifica?
- Il server invia la risposta: Client, NO
- la connessione finisce

2. Nuova richiesta :

- Il cliente invia una richiesta: Server, hai una nuova notifica?
- Il server invia la risposta: Client, NO
- la connessione finisce

3. Nuova richiesta :

- Il cliente invia una richiesta: Server, hai una nuova notifica?
- Il server invia la risposta: Client, NO
- la connessione finisce

4. Nuova richiesta :

- Il cliente invia una richiesta: Server, hai una nuova notifica?
- Il server invia la risposta: Client, Sì, eccoti qui
- la connessione finisce

vedi che c'è un numero di richieste inutili!

web-zoccolo

- connessione stabilita :
- il client dice: Server, hai una nuova notifica?

dopo che xx è passato

- Server dice: Sì, ho

dopo che xx è passato

- Server dice: Sì, ho
- Il cliente dice: Server, invia una nuova email se ho

dopo che xx è passato

- Il server dice: ora hai una nuova email

ora è chiaro che con la web-socket non avremo richieste inutili

Leggi [Differenze tra WebSockets e AJAX online](https://riptutorial.com/it/websocket/topic/8139/differenze-tra-websockets-e-ajax):

<https://riptutorial.com/it/websocket/topic/8139/differenze-tra-websockets-e-ajax>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con WebSockets	Brock Adams , Community , Farhad
2	Differenze tra WebSockets e AJAX	Farhad , jalanb