

MOBILE DEVELOPER

The background is a solid blue color with a faint, light-blue network diagram. This diagram consists of numerous small circular nodes connected by thin, light-blue lines, creating a complex web-like structure. A large, thin white arc is positioned in the upper right quadrant of the image. In the bottom right corner, there is a small, solid dark-blue circular shape.

Le Web socket

Sommario

- Le web socket.

In passato

nel corso degli anni, la comunicazione fra client server con modalità full-duplex avveniva:

socket in Flash pilotati via Javascript

la tecnica di long polling (l'utilizzo continuo di chiamate AJAX mantenute aperte fino alla ricezione del dato o al tempo di timeout).

Web Socket: Introduzione

Le web socket rappresentano uno dei modi di comunicazioni più potenti offerti da HTML 5

Una *socket* è un canale di comunicazione full-duplex basato sui protocolli *sw* o *ssw*.

Web Socket: obiettivo

Le Web Socket API introducono, nella loro estrema semplicità, una funzionalità tra le più attese ed emulate: la possibilità di stabilire e mantenere una connessione dati tra browser e server remoto. In questo canale di comunicazione si fanno transitare messaggi in entrambe le direzioni.

Web Socket: entità principali

In buona sostanza le API offrono un meccanismo molto semplice grazie a:

oggetto *WebSocket*.

metodo *send*.

evento *onMessage*.

Web Socket: eventi

Gli eventi salienti di una Web Socket sono essenzialmente i seguenti:

open: crea la connessione.

message: riceve il messaggio.

close: chiude la connessione.

error: gestione dell'errore.

Web Socket: protocollo

Le comunicazioni viaggiano all'interno di un nuovo e specifico protocollo, chiamato 'The WebSocket Protocol' che richiama per certi aspetti, soprattutto durante l'handshake, una conversazione HTTP.

Web Socket: creazione

La creazione di un nuovo WebSocket richiede come unico parametro obbligatorio l'url verso la quale si vuole stabilire la connessione. Il protocollo può essere ws o wss. Opzionalmente è possibile passare al costruttore anche una stringa o un array di sub-protocolli.

```
var echo_service = new WebSocket('ws://echo.websocket.org');
```

Web Socket: onMessage

Una volta creato un nuovo WebSocket, il funzionamento dello stesso diventa praticamente identico, nella forma, a quello già esposto per la comunicazione tra Worker: la funzione associata all'handler onMessage che viene invocata ogniqualvolta dal server proviene un messaggio.

```
echo_service.onmessage = function(event){  
    alert(event.data);  
}
```

Web Worker: onOpen

La funzione `send` provvede all'invio, verso il server remoto, del testo passato come argomento. Da notare che l'invio deve essere necessariamente subordinato alla condizione di avvenuta connessione, notificata tramite l'evento `onOpen`.

```
echo_service.onopen = function(){  
  echo_service.send("hello!");  
}
```

Web Socket: Server

E' immediato intuire che per funzionare le WebSocket hanno bisogno anche de lato server, ovvero di un software che sappia rispondere e gestire le comunicazioni lato client. Alcune soluzioni attualmente disponibili sono:

Socket.IO (Java e node.js)

node-websocket-server

Jetty (Java)

Web Socket: riferimenti

Giunti al termine dell'argomento Web Socket, di seguito si riportano riferimenti per approfondire l'argomento:

Il sito "ufficiale": <http://websocket.org>.

La specifica delle API (client):
<http://dev.w3.org/html5/websockets/>.

La specifica del protocollo (server):
<http://tools.ietf.org/html/draft-ietf-hybi-thewebsocketprotocol-03>.

Domande & approfondimenti

MOBILE DEVELOPER