

HTML5 - WEBSOCKETS

Web Sockets is a next-generation bidirectional communication technology for web applications which operates over a single socket and is exposed via a JavaScript interface in HTML 5 compliant browsers.

Once you get a Web Socket connection with the web server, you can send data from browser to server by calling a **send** method, and receive data from server to browser by an **onmessage** event handler.

Following is the API which creates a new WebSocket object.

```
var Socket = new WebSocket(url, [protocol] );
```

Here first argument, url, specifies the URL to which to connect. The second attribute, protocol is optional, and if present, specifies a sub-protocol that the server must support for the connection to be successful.

WebSocket Attributes

Following are the attribute of WebSocket object. Assuming we created Socket object as mentioned above –

Attribute	Description
Socket.readyState	<p>The readonly attribute readyState represents the state of the connection. It can have the following values –</p> <ul style="list-style-type: none">• A value of 0 indicates that the connection has not yet been established.• A value of 1 indicates that the connection is established and communication is possible.• A value of 2 indicates that the connection is going through the closing handshake.• A value of 3 indicates that the connection has been closed or could not be opened.
Socket.bufferedAmount	<p>The readonly attribute bufferedAmount represents the number of bytes of UTF-8 text that have been queued using send method.</p>

WebSocket Events

Following are the events associated with WebSocket object. Assuming we created Socket object as mentioned above –

Event	Event Handler	Description
open	Socket.onopen	This event occurs when socket connection is established.
message	Socket.onmessage	This event occurs when client receives data from server.
error	Socket.onerror	This event occurs when there is any error in

communication.

close Socket.onclose This event occurs when connection is closed.

WebSocket Methods

Following are the methods associated with WebSocket object. Assuming we created Socket object as mentioned above –

Method	Description
Socket.send	The <code>send</code> method transmits data using the connection.
Socket.close	The <code>close</code> method would be used to terminate any existing connection.

WebSocket Example

A WebSocket is a standard bidirectional TCP socket between the client and the server. The socket starts out as a HTTP connection and then "Upgrades" to a TCP socket after a HTTP handshake. After the handshake, either side can send data.

Client Side HTML & JavaScript Code

At the time of writing this tutorial, there are only few web browsers supporting WebSocket interface. You can try following example with latest version of Chrome, Mozilla, Opera and Safari.

```
<!DOCTYPE HTML>
<html>
  <head>

    <script type="text/javascript">
      function WebSocketTest()
      {
        if ("WebSocket" in window)
        {
          alert("WebSocket is supported by your Browser!");

          // Let us open a web socket
          var ws = new WebSocket("ws://localhost:9998/echo");

          ws.onopen = function()
          {
            // Web Socket is connected, send data using send()
            ws.send("Message to send");
            alert("Message is sent...");
          };

          ws.onmessage = function (evt)
          {
            var received_msg = evt.data;
            alert("Message is received...");
          };

          ws.onclose = function()
          {
            // websocket is closed.
            alert("Connection is closed...");
          };
        }
      }
    </script>
  </head>
</html>
```

```

        else
        {
            // The browser doesn't support WebSocket
            alert("WebSocket NOT supported by your Browser!");
        }
    }
</script>

</head>
<body>

    <div >
        <a href="javascript:WebSocketTest()">Run WebSocket</a>
    </div>

</body>
</html>

```

Install pywebsocket

Before you test above client program, you need a server which supports WebSocket. Download `mod_pywebsocket-x.x.x.tar.gz` from [pywebsocket](#) which aims to provide a Web Socket extension for Apache HTTP Server and install it following these steps.

- Unzip and untar the downloaded file.
- Go inside **pywebsocket-x.x.x/src/** directory.
- `$python setup.py build`
- `$sudo python setup.py install`
- Then read document by:
 - `$pydoc mod_pywebsocket`

This will install it into your python environment.

Start the Server

Go to the **pywebsocket-x.x.x/src/mod_pywebsocket** folder and run the following command –

```
$sudo python standalone.py -p 9998 -w ../example/
```

This will start the server listening at port 9998 and use the handlers directory specified by the `-w` option where our `echo_wsh.py` resides.

Now using Chrome browser open the html file you created in the beginning. If your browser supports WebSocket, then you would get alert indicating that your browser supports WebSocket and finally when you click on "Run WebSocket" you would get Goodbye message sent by the server script.

```
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```