

WebSockets - Events & Actions

It is necessary to initialize the connection to the server from client for communication between them. For initializing the connection, creation of Javascript object with the URL with the remote or local server is required.

```
var socket = new WebSocket(" ws://echo.websocket.org ");
```

The URL mentioned above is a public address that can be used for testing and experiments. The websocket.org server is always up and when it receives the message and sends it back to the client.

This is the most important step to ensure that application works correctly.

Web Sockets – Events

There are four main Web Socket API **events** –

- Open
- Message
- Close
- Error

Each of the events are handled by implementing the functions like **onopen**, **onmessage**, **onclose** and **onerror** functions respectively. It can also be implemented with the help of addEventListener method.

The brief overview of the events and functions are described as follows –

Open

Once the connection has been established between the client and the server, the open event is fired from Web Socket instance. It is called as the initial handshake between client and server. The event, which is raised once the connection is established, is called **onopen**.

Message

Message event happens usually when the server sends some data. Messages sent by the server to the client can include plain text messages, binary data or images. Whenever the data is sent, the **onmessage** function is fired.

Close

Close event marks the end of the communication between server and the client. Closing the connection is possible with the help of **onclose** event. After marking the end of communication with the help of **onclose** event, no messages can be further transferred between the server and the client. Closing the event can happen due to poor connectivity as well.

Error

Error marks for some mistake, which happens during the communication. It is marked with the help of **onerror** event. **Onerror** is always followed by termination of connection. The detailed description of each and every event is discussed in further chapters.

Web Sockets – Actions

Events are usually triggered when something happens. On the other hand, actions are taken when a user wants something to happen. Actions are made by explicit calls using functions by users.

The Web Socket protocol supports two main actions, namely –

- `send()`
- `close()`

`send ()`

This action is usually preferred for some communication with the server, which includes sending messages, which includes text files, binary data or images.

A chat message, which is sent with the help of `send()` action, is as follows –

```
// get text view and button for submitting the message
var textsend = document.getElementById("text-view");
var submitMsg = document.getElementById("tsend-button");

//Handling the click event
submitMsg.onclick = function ( ) {
    // Send the data
    socket.send( textsend.value);
}
```

Note – Sending the messages is only possible if the connection is open.

`close ()`

This method stands for goodbye handshake. It terminates the connection completely and no data can be transferred until the connection is re-established.

```
var textsend = document.getElementById("text-view");
var buttonStop = document.getElementById("stop-button");
```

```
//Handling the click event
buttonStop.onclick = function ( ) {
    // Close the connection if open
    if (socket.readyState === WebSocket.OPEN){
        socket.close( );
    }
}
```

It is also possible to close the connection deliberately with the help of following code snippet –

```
socket.close(1000,"Deliberate Connection");
```