Chapter 61: WebSockets

Parameter	Details
url	The server url supporting this web socket connection.
data	The content to send to the host.
message	The message received from the host.

WebSocket is protocol, which enables two-way communication between a client and server:

The goal WebSocket is to provide a mechanism for browser-based applications that need two-way communication with servers that does not rely on opening multiple HTTP connections. (RFC 6455)

WebSocket works over HTTP protocol.

Section 61.1: Working with string messages

```
var wsHost = "ws://my-sites-url.com/path/to/echo-web-socket-handler";
var ws = new WebSocket(wsHost);
var value = "an example message";

//onmessage : Event Listener - Triggered when we receive message form server
ws.onmessage = function(message) {
    if (message === value) {
        console.log("The echo host sent the correct message.");
    } else {
        console.log("Expected: " + value);
        console.log("Received: " + message);
    }
};

//onopen : Event Listener - event is triggered when websockets readyState changes to open which means
now we are ready to send and receives messages from server
ws.onopen = function() {
    //send is used to send the message to server
    ws.send(value);
};
```

Section 61.2: Establish a web socket connection

```
var wsHost = "ws://my-sites-url.com/path/to/web-socket-handler";
var ws = new WebSocket(wsHost);
```

Section 61.3: Working with binary messages

```
var wsHost = "http://my-sites-url.com/path/to/echo-web-socket-handler";
var ws = new WebSocket(wsHost);
var buffer = new ArrayBuffer(5); // 5 byte buffer
var bufferView = new DataView(buffer);

bufferView.setFloat32(0, Math.PI);
bufferView.setUint8(4, 127);

ws.binaryType = 'arraybuffer';

ws.onmessage = function(message) {
    var view = new DataView(message.data);
```

```
    console.log('Uint8:', view.getUint8(4), 'Float32:', view.getFloat32(0))
};

ws.onopen = function() {
    ws.send(buffer);
};
```

Section 61.4: Making a secure web socket connection

```
var sck = "wss://site.com/wss-handler";
var wss = new WebSocket(sck);
```

This uses the `wss` instead of `ws` to make a secure web socket connection which make use of HTTPS instead of HTTP