

The following are questions I asked ChatGPT for lab 1 and it's responses.

How to I make the messages and status stay on the html page. They flash briefly

ChatGPT

ChatGPT

To make the messages and status stay on the HTML page instead of flashing briefly, you need to prevent the default form submission behavior. This behavior causes the page to reload, clearing out any changes made by JavaScript. You can achieve this by modifying the event handlers to accept the event parameter and calling the preventDefault() method on it.

Here's the updated code with the necessary changes:

From <<https://chat.openai.com/c/9158e69b-1450-445c-a25e-d0d35d51afb5>>

B

You

```
<!-- Form for MQTT connection --> <form id="connform"> Server: <input type="text"
name="server" value="broker.emqx.io"><br> Port: <input type="text" name="port"
value="8084"><br> <button onclick="MQTTconnect()">Start</button> <button
onclick="endConnection()">End</button> </form> How to hide the server and port form if I
click the start button and only show them again once I hit end
```

ChatGPT

ChatGPT

You can achieve this functionality using JavaScript by manipulating the CSS display property of the form elements. Here's how you can modify your HTML and JavaScript to hide and show the server and port inputs based on the button clicks:

HTML:

htmlCopy code

```
<!DOCTYPE
html><html lang="en"><head><meta charset="UTF-8"><meta name="viewport" content="width
=device-width, initial-scale=1.0"><title>MQTT WebSocket Example</title></head><body><!--
Form for MQTT connection --><form id="connform">Server:
<input type="text" name="server" value="broker.emqx.io" id="serverInput"><br>Port:
<input type="text" name="port" value="8084" id="portInput"><br><button onclick="startConnec
tion()">Start</button><button onclick="endConnection()">End</button></form><!-- Include
the MQTT WebSocket library
--><script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js"></scr
ipt><!-- Include your custom JavaScript code --><script src="script.js"></script></body></html>
JavaScript (script.js):
```

javascriptCopy code

```
// Function to start MQTT connectionfunction startConnection() { // Hide server and port
inputsdocument.getElementById("serverInput").style.display = "none";
```

```

document.getElementById("portInput").style.display= "none"; // Call MQTT connect
functionMQTTConnect(); } // Function to end MQTT connectionfunctionendConnection() { //
Show server and port inputsdocument.getElementById("serverInput").style.display= "inline";
document.getElementById("portInput").style.display= "inline"; // Call MQTT disconnect
function if necessary// MQTTdisconnect();} // Function to connect to MQTT
brokerfunctionMQTTConnect() { // Get server and port valuesvarserver =
document.getElementById("serverInput").value; varport =
document.getElementById("portInput").value; // MQTT broker URLvarmqttServer = "wss://" +
server + ":" + port + "/mqtt"; // Create a client instancevarclient =
newPaho.MQTT.Client(mqttServer, "clientId_" + Math.random().toString(16).substr(2, 8)); // Set
callback handlersclient.onConnectionLost= onConnectionLost; client.onMessageArrived=
onMessageArrived; // Connect the clientclient.connect({ onSuccess: onConnect, onFailure:
onFailure, useSSL: true// Use SSL/TLS for secure WebSocket connection}); } // Called when the
connection is successfulfunctiononConnect() { console.log("Connected to MQTT broker"); //
Perform any actions upon successful connection} // Called when the connection is
lostfunctiononConnectionLost(responseObject) { if(responseObject.errorCode!== 0) {
console.log("Connection lost: " + responseObject.errorMessage); // Attempt to reconnect or
handle the connection loss} } // Called when a message
arrivesfunctiononMessageArrived(message) { console.log("Message received: " +
message.payloadString); // Process the received message} // Called when the connection
attempt failsfunctiononFailure(errorMessage) { console.log("Connection failed: " +
errorMessage); // Handle the connection failure}

```

With this implementation, when you click the "Start" button, the server and port inputs will be hidden, and the MQTT connection process will start. When you click the "End" button, the server and port inputs will be shown again.

From <<https://chat.openai.com/c/bed6926a-de0e-4c01-b96d-e14bb54be518>>

```

function onConnect() { console.log("Connected to " + host + " on port " + port);
document.getElementById("messages").innerHTML = "Connected to " + host + " on port " +
port; document.getElementById("status").innerHTML = "Connected"; connected_flag = true; }
How do I make it automatically subscribe?

```

ChatGPT

ChatGPT

To automatically subscribe to a topic once the MQTT client is connected, you can add the subscription logic inside the onConnect function after establishing the connection. Here's how you can modify your onConnect function to subscribe to a specific topic:

javascriptCopy code

```

functiononConnect() { console.log("Connected to " + host + " on port " + port);
document.getElementById("messages").innerHTML= "Connected to " + host + " on port " + port;

```

```
document.getElementById("status").innerHTML= "Connected"; connected_flag = true; //
Subscribe to a specific topicclient.subscribe("your/topic", { qos: 1}); // Replace "your/topic"
with the topic you want to subscribe to}
```

In the above code:

- Replace "your/topic" with the topic you want to subscribe to.
- { qos: 1 } specifies the Quality of Service (QoS) level for the subscription. You can adjust it as per your requirements.

By adding the subscription logic inside the onConnect function, the MQTT client will automatically subscribe to the specified topic once the connection is established successfully.

From <<https://chat.openai.com/c/bed6926a-de0e-4c01-b96d-e14bb54be518>>

```
function success(position) { status.textContent = "Location found:" latitude.textContent =
position.coords.latitude; longitude.textContent = position.coords.longitude;
client.subscribe("ENG0551/braden_navarro/my_temperature", { qos: 1 }); } So how would I send
the lat and long but also subscribe to it to get those values from another function?
```

ChatGPT

ChatGPT

To send the latitude and longitude coordinates while also subscribing to receive them in another function, you can follow these steps:

1. Publish Latitude and Longitude: Within the success function, after retrieving the latitude and longitude coordinates, you can publish them to a specific MQTT topic.
2. Subscribe to the Topic: In another function, such as subscribeToLocation, you can subscribe to the same MQTT topic where the latitude and longitude coordinates are being published.

Here's how you can modify your code to achieve this:

javascriptCopy code

```
// Function to publish latitude and longitude coordinatesfunctionpublishLocation(latitude,
longitude) { vartopic = "ENG0551/braden_navarro/my_location"; varmessage = JSON.stringify({
latitude: latitude, longitude: longitude }); varmessageObj = newPaho.MQTT.Message(message);
messageObj.destinationName= topic; client.send(messageObj); }
```

From <<https://chat.openai.com/c/bed6926a-de0e-4c01-b96d-e14bb54be518>>

Yes, you can generate random values within a specific range using JavaScript's Math.random() function. Here's how you can modify your code to generate a random client ID within a range:

javascriptCopy code

```
// Generate a random client ID within a specific rangevar randomClientId = "clientID_" +  
Math.floor(Math.random() * (max - min + 1)) + min; // Create MQTT client with the generated  
random client IDmqtt = newPaho.MQTT.Client(host, port, randomClientId);
```

In this code:

- max and min represent the upper and lower bounds of the range you want to generate random numbers within.
- Math.random() generates a floating-point number between 0 (inclusive) and 1 (exclusive).
- Math.floor() rounds down to the nearest integer.
- (max - min + 1) calculates the range of possible values.
- Adding min to the result ensures that the random number falls within the desired range.

For example, if you want to generate random client IDs between 1 and 100, you would set min to 1 and max to 100:

javascriptCopy code

```
var min = 1; var max = 100; var randomClientId = "clientID_" + Math.floor(Math.random() * (max  
- min + 1)) + min; mqtt = newPaho.MQTT.Client(host, port, randomClientId);
```

This will create a random client ID like "clientID\_42", "clientID\_73", etc.

From <<https://chat.openai.com/c/bed6926a-de0e-4c01-b96d-e14bb54be518>>