

ENGO 585 Lab 5 Sample ChatGPT Prompts

Mitchell Aitken - 2023-03-09

Query

when it sends in these coordinates, what is the format in this message? Topic:

ENGO551/Mitchell/my_temperatureQoS: 0

```
{"type": "Feature", "properties": {"temperature": -37.694855889926345}, "geometry": {"type": "Point", "coordinates": [-114.2325248, 51.1049728]}}
```

ChatGPT

In the MQTT message you've provided:

json

Copy code

```
{
  "type": "Feature",
  "properties": {
    "temperature": -37.694855889926345
  },
  "geometry": {
    "type": "Point",
    "coordinates": [-114.2325248, 51.1049728]
  }
}
```

The format of the coordinates is typical of GeoJSON, where the coordinates property is an array with two or more elements:

The first element is the longitude (-114.2325248).

The second element is the latitude (51.1049728).

This ordering of longitude first, followed by latitude, is the standard for GeoJSON, but it is the opposite of what many mapping libraries (including Leaflet) expect, which is latitude followed by longitude.

So when you use these coordinates in Leaflet, you need to reverse them:

javascript

Copy code

```
var longitude = receivedData.geometry.coordinates[0];
var latitude = receivedData.geometry.coordinates[1];
currentLocationMarker.setLatLng([latitude, longitude]);
```

This will place the marker at the correct location based on the GeoJSON coordinates provided by the MQTT message.

Query

I still get this error: Uncaught DOMException: Failed to construct 'WebSocket': The URL

'wss://[wss://broker.emqx.io]:8084/mqtt' is invalid.

at k._doConnect (<https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js:36:453>)

at k.connect (<https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js:31:481>)

at l.connect (<https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js:65:82>)

at HTMLButtonElement.connectToMQTT (<https://mitchell-aitken.github.io/ENGO551Lab5/app.js:11:12>)

2mqttws31.min.js:34 Uncaught Error: AMQJS0011E Invalid state not connected.

at k.send (mqttws31.min.js:34:175)

at l.send (mqttws31.min.js:67:296)

at app.js:84:16

mqttws31.min.js:36 Uncaught DOMException: Failed to construct 'WebSocket': The URL

'wss://[wss://broker.emqx.io]:8084/mqtt' is invalid.

at k._doConnect (https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js:36:453)

at k.connect (https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js:31:481)

at l.connect (https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js:65:82)

at HTMLButtonElement.connectToMQTT (https://mitchell-aitken.github.io/ENGO551Lab5/app.js:11:12)...

This is my code. var client; // Here's our MQTT client

var map; // This guy will hold our map

var currentLocationMarker; // This marker shows where we are right now

ChatGPT

The error you're encountering, Uncaught DOMException: Failed to construct 'WebSocket': The URL

'wss://[wss://broker.emqx.io]:8084/mqtt' is invalid., suggests that there's a problem with how the WebSocket URL is being constructed. The URL should not have the scheme (wss://) repeated or enclosed in brackets, and it should not have /mqtt at the end unless specified by your MQTT broker.

Here's how you can correct the URL construction in your connectToMQTT function:

Correct the WebSocket URL Formation:

The MQTT client expects just the hostname or IP address and port, not a full URL. The library will handle the WebSocket protocol (ws:// or wss://) part based on whether you use SSL.

Remove Redundant Code:

Since you're already using 'wss://' (WebSocket Secure), there's no need to prepend 'wss://' again to the host.

Query

I tried changing the markers with images of different colours but that doesn't seem to be working, can you use html to do this instead?

ChatGPT

To address the issue of changing the marker's color without needing an image file, you can modify the createCustomIcon function. Here's an example using Leaflet's L.divIcon to create a marker with a custom background color:

javascript

Copy code

```
function createCustomIcon(color) {
  return L.divIcon({
    className: 'custom-color-marker',
    html: "<div style='background-color: " + color + "; width: 10px; height: 10px; border-radius: 50%;'></div>",
    iconSize: [10, 10],
    iconAnchor: [5, 5]
  });
}
```

This function creates a simple circular div icon whose color can be set dynamically. Adjust the size, shape, and style as needed for your application.

Query

My code isn't allowing the markers to display properly even though the coordinates are being sent through MQTT. Do I need to reorder my functions in my java code?
(pasted code)

ChatGPT

Reviewing your provided JavaScript code for the MQTT and Leaflet map setup, the functions are organized correctly in terms of their definitions and the flow of logic. However, the issue seems to be related to the `onMessageArrived` function not updating the map's position, even though the correct coordinates are being received from the MQTT message.

The structure of your code and the order of function definitions typically do not impact the functionality in this context, as JavaScript function declarations are hoisted. However, the execution flow and the way functions are called matter significantly. Here's a closer look:

MQTT Subscription: You're correctly subscribing to the "ENG0551/Mitchell/my_temperature" topic upon successfully connecting to the MQTT broker. This part seems to be working fine since you're seeing the successful connection message.

Message Arrival Handling: Your `onMessageArrived` function is designed to update the marker's position on the map based on the incoming MQTT message. However, since you're only seeing the connection success message and not the logs from `onMessageArrived`, it suggests that either the messages aren't arriving at your client, or they are not triggering this function.

Marker and Map Update: The code intended to update the marker's position and the map's view seems logically correct. However, if the `onMessageArrived` function is not being triggered or not working as expected, the map will remain at the initial coordinates `[0, 0]`.

To troubleshoot further:

Check MQTT Message Flow: Use an MQTT client like MQTT Explorer or MQTT.fx to subscribe to the same topic and verify that the messages are being published correctly.

Debugging: Add more `console.log` statements in the `onMessageArrived` function to ensure it's being triggered and to log the received data. If this function is not logging anything, the issue could be with the MQTT subscription or message delivery.

MQTT Client Configuration: Ensure that `client.onMessageArrived = onMessageArrived;` is correctly set up before calling `client.connect(...)`. This ensures that the MQTT client knows which function to call when a new message arrives.

Map Initialization: Consider initializing your map with a more realistic default location or updating the map only after receiving the first valid MQTT message.

Error Handling: Check the browser's console for any errors or warnings that might indicate what's going wrong.

Testing the Function Manually: Temporarily simulate the `onMessageArrived` function call by manually invoking it with a mock message object to see if the map updates as expected.

Regarding the function order, JavaScript hoists the function declarations to the top of their scope before code execution, meaning the order of function definitions in your file should not be the issue. The critical part is ensuring the functions are correctly referenced and used, especially the callback setups for the MQTT client.

If after these checks you're still facing issues, consider simplifying the code or isolating the problem by manually calling `onMessageArrived` with hardcoded data to ensure the map updating logic works independently of the MQTT message reception.