

SIT314 Software Architecture And Scalability For Internet-Of-Things

Pass Task 8.2: Publish and Subscribe to the MQTT Broker

Task Overview

In the previous task, you have learned how to publish/subscribe with `Node.js` from the command line via an public MQTT broker provided by `HiveMQTT`. In this task, you will extend the code in 8.1P and implement a simple server and client script to publish/subscribe topics.

We continue the 'Rent-a-Drone' scenario. In this task, you will implement a publisher to send data from your drones (client), and a subscriber to receive data from the owner's end (server).

Submission Details

Submit the following as a single document to [OnTrack](#):

- `server.js` and `client.js` code
- Screenshot of your console showing your published and subscribed messages

Instructions

Publisher - `client.js`

1. To use the client, you will first need to use the following at the top of your script:

```
const mqtt = require('mqtt')
```

Note: You may need to reinstall MQTT with the command:

```
sudo npm install mqtt@4.0.1 --save
```

2. Node.js scripts are asynchronous and use events. Events are triggered when something happens e.g the client connects. You create a listener for the events you are interested in and this listener calls a callback function to process the event. Now we create the connection method as follows:

```
const client = mqtt.connect("mqtt://broker.hivemq.com:1883");
```

3. Now let's create two variables - `topic` and `message`:

```
var topic="/myTopic"
var message="My message"
```

4. The MQTT protocol acknowledges a connection with the CONNACK message. This raises the `on_connect` event in the client which can be examined by creating a listener. We create the listening and publish the `/myTopic` topic as follows:

```
client.on('connect', () =>
{
  console.log('mqtt connected');
  client.publish(topic, message);
  console.log('published to Topic: ' + topic + " with Message: " +
message);
});
```

Subscriber - server.js

1. Now we would like to subscribe to the topics and messages published by `client.js`
2. Create a new script called `server.js`. You will need to connect to the `HiveMQTT` broker like what you have done in `client.js`.

```
const mqtt = require('mqtt')
const client = mqtt.connect("mqtt://broker.hivemq.com:1883");
```

3. Now set the topic variable to the one you have created at the client:

```
var topic="/myTopic"
```

Note: You use a topic string method if you only want to subscribe to a single topic. You will need to use the array method if you want to subscribe to multiple topics with the same `qos`. You will need to use the object method if you want to subscribe to multiple topics with different `qos` settings.

4. Before you can receive any messages you will need to be subscribed to a topic. The message event is triggered when a new message arrives. To process this event you need to create a listener as follows:

```
client.on('connect', () =>
{
  client.subscribe(topic);
  console.log('mqtt connected');
});

client.on('message', (topic, message) =>
```

```
{  
  console.log("Topic is: " + topic)  
  console.log("Message is: " + message)  
});
```

5. You should now be able to see the subscribed messages printed to the console!