

Lab work #3 MQTT app paho

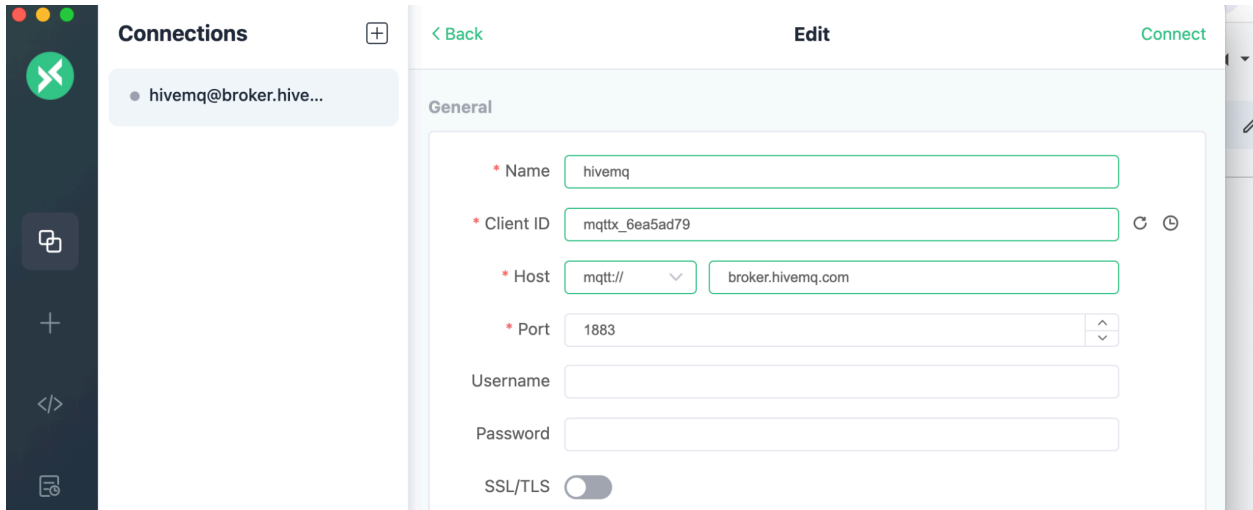
Task: create a basic MQTT app, that would be running using paho mqtt python package

Examples:

<https://medium.com/python-point/mqtt-basics-with-python-examples-7c758e605d4>

Workflow:

1. Install MQTTx app <https://mqttx.app/downloads>
2. Add MQTTx client correct MQTT broker server details (below)



The screenshot shows the MQTTx application interface. On the left is a dark sidebar with icons for connections, a plus sign, and code editor. The main area is titled 'Connections' and shows a list of connections, including 'hivemq@broker.hive...'. To the right, the 'Edit' screen for this connection is displayed. It has a 'General' tab with the following fields:

- Name:** hivemq
- Client ID:** mqttx_6ea5ad79
- Host:** mqtt:// broker.hivemq.com
- Port:** 1883
- Username:** (empty field)
- Password:** (empty field)
- SSL/TLS:** (disabled toggle switch)

At the top right of the 'Edit' screen are buttons for '< Back' and 'Connect'.

3. Create mqtt python code, below:

```
1  import paho.mqtt.client as mqtt
2  import time
3
4  def connect_broker(broker_address, client_name):
5      client = mqtt.Client(client_name)
6      client.connect(broker_address)
7      time.sleep(1)
8      client.loop_start()
9
10     return client
11
12  if __name__ == "__main__":
13      server = "broker.hivemq.com"
14      client_name = "tester"
15      client = connect_broker(server, client_name)
16      try:
17          while True:
18              message = input('Send some random message:')
19              client.publish("temperature",message)
20      except KeyboardInterrupt:
21          client.disconnect()
22          client.loop_stop()
23
```

4. In similar way try to add subscriber option instead of publishing so you would be able to receive the messages from MQTTx that you are sending

The screenshot shows the hivemq MQTT client interface. On the left, there is a sidebar with a 'hivemq' logo, a status indicator (143), and a '+ New Subscription' button. Below this, a subscription is listed for the 'temperature' topic with QoS 0. The main panel displays a list of received messages for the 'temperature' topic. The messages are shown in a list with a 'Plaintext' filter selected. The messages are as follows:

- Topic: temperature QoS: 0
{
 "msg": "33"
}
2023-11-23 13:21:35:983
- Topic: temperature QoS: 0
{
 "msg": "33"
}
2023-11-23 13:21:40:684
- Topic: temperature QoS: 0
{
 "msg": "33"
}
2023-11-23 13:21:40:723

At the bottom, there is a detailed view of the selected message, showing the topic 'temperature' and the JSON payload: { "msg": "33" }. The interface also includes a 'JSON' filter, 'QoS 0' filter, and a 'Retain' checkbox.