

Develop the Python Script

(Publish data to IBM cloud)

Date	04 November 2022
Team ID	PNT2022TMID13566
Project Name	Industry-specific intelligent fire management system
Maximum Marks	4 Marks

Industry-specific intelligent fire management system



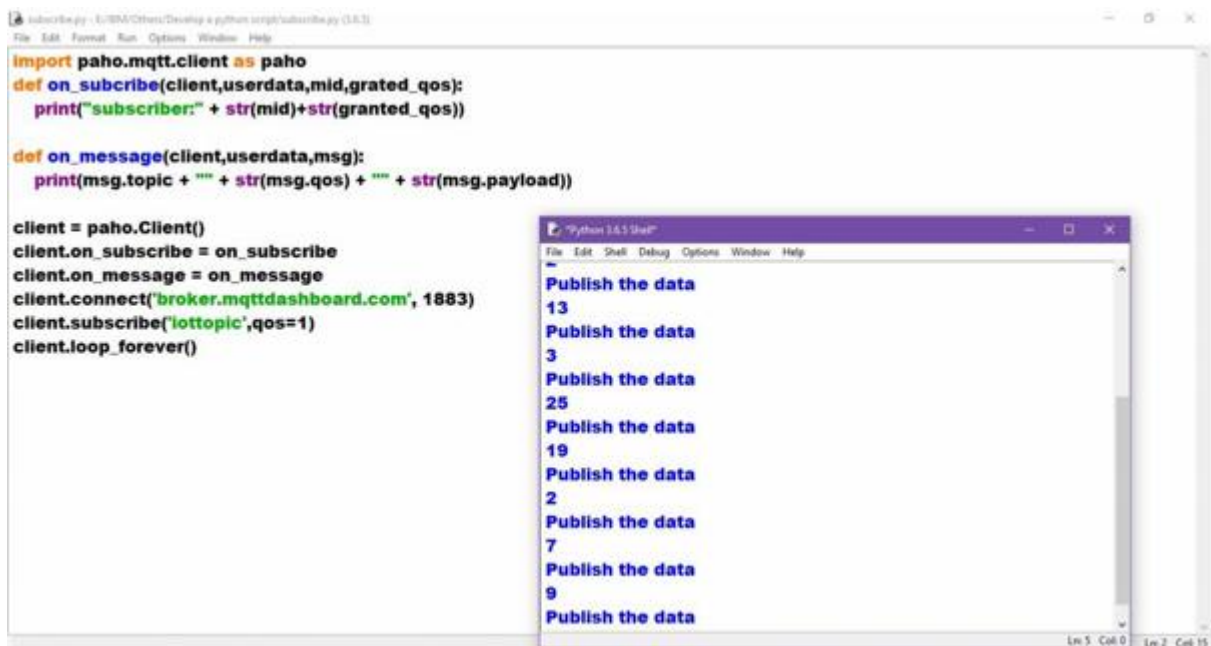
The screenshot shows a Python script in a text editor and its execution output in a terminal window. The script, named `publish.py`, is designed to publish data to an MQTT broker. It imports `paho.mqtt.client` as `paho`, `time`, and `random`. It defines a function `on_publish` that prints "Publish the data". The main logic creates a `paho.Client`, connects to `broker.mqttdashboard.com` on port 1883, and enters a `while True` loop. In each iteration, it generates a random integer between 1 and 30, publishes it to the `iottopic` with a QoS of 1, prints the value, and sleeps for 10 seconds.

```
#Through python coding we are going to access the subscriber
import paho.mqtt.client as paho
import time
import random

def on_publish(client, userdata, mid):
    print("Publish the data ")

client = paho.Client()
client.on_publish = on_publish
client.connect('broker.mqttdashboard.com', 1883)
client.loop_start()
while True:
    temp = random.randint(1,30)
    (re,mid) = client.publish('iottopic',str(temp),qos=1)
    print(temp)
    time.sleep(10)
```

The terminal window shows the Python version (3.6.5) and the output of the script, which is a series of numbers: 7, 19, 10, and 10, each preceded by "Publish the data".



The screenshot shows a Python script in a text editor and its execution output in a terminal window. The script, named `subscribe.py`, is designed to subscribe to an MQTT broker. It imports `paho.mqtt.client` as `paho`. It defines two functions: `on_subscribe`, which prints the subscription status, and `on_message`, which prints the received message. The main logic creates a `paho.Client`, connects to `broker.mqttdashboard.com` on port 1883, subscribes to the `iottopic` with a QoS of 1, and enters a `client.loop_forever()` loop.

```
import paho.mqtt.client as paho
def on_subscribe(client,userdata,mid,grated_qos):
    print("subscriber:" + str(mid)+str(grated_qos))

def on_message(client,userdata,msg):
    print(msg.topic + "" + str(msg.qos) + "" + str(msg.payload))

client = paho.Client()
client.on_subscribe = on_subscribe
client.on_message = on_message
client.connect('broker.mqttdashboard.com', 1883)
client.subscribe('iottopic',qos=1)
client.loop_forever()
```

The terminal window shows the output of the script, which is a series of messages: "Publish the data", "13", "Publish the data", "3", "Publish the data", "25", "Publish the data", "19", "Publish the data", "2", "Publish the data", "7", "Publish the data", "9", and "Publish the data".

IBM Watson IoT Platform

mathavanvj15@gmail.com
ID: pq685h

Device ID: abcd, Status: Disconnected, Device Type: 123, Date Added: Nov 3, 2022 12:13 PM

Identity | Device Information | **Recent Events** | State | Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{\"randomNumber\":74}	json	a few seconds ago
event_1	{\"randomNumber\":47}	json	a few seconds ago
event_1	{\"randomNumber\":45}	json	a minute ago
event_1	{\"randomNumber\":19}	json	a minute ago
event_1	{\"randomNumber\":79}	json	a minute ago

1 Simulation running

IBM Watson IoT Platform

mathavanvj15@gmail.com
ID: (select org)

Buildings

Collect data from

and make value from it

Learn More

Program :

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
```

```

myConfig = {"identity":
{
    "orgId": "hj5fmy",
    "typeId": "NodeMCU",
    "deviceId": "12345" },
    "auth": { "token": "12345678" }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
    client.disconnect()

```