

Develop the Python Script

(Publish data to IBM cloud)

Date	November 2022
Team ID	PNT2022TMID33937
Project Name	Industry-Specific Intelligent Fire Management System
Maximum Marks	4 Marks

The screenshot shows a Python script named 'publish.py' in a text editor. The script imports 'paho.mqtt.client' as 'paho', 'time', and 'random'. It defines a function 'on_publish' that prints 'Publish the data'. The main code creates a 'paho.Client', sets 'on_publish' as the callback, connects to 'broker.mqttdashboard.com' on port 1883, and starts the loop. A 'while True' loop publishes random data (1-30) to the 'iottopic' every 10 seconds.

Next to the script is a terminal window titled 'Python 3.6.5 Shell'. It shows the command prompt, the file path, and the output of the script: 'Publish the data' followed by the numbers 19, 10, and 10.

```
#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)
```

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MS
C v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more informati
n.
>>>
===== RESTART: E:\IBM\Others\Develop a python script\
publish.py =====
7
Publish the data
19
Publish the data
10
Publish the data
```

The screenshot shows a Python script named 'subscribe.py' in a text editor. It imports 'paho.mqtt.client' as 'paho'. It defines two functions: 'on_subscribe' which prints the subscriber ID and granted QoS, and 'on_message' which prints the message topic, QoS, and payload. The main code creates a 'paho.Client', sets the callbacks, connects to 'broker.mqttdashboard.com' on port 1883, subscribes to 'iottopic' with QoS 1, and starts the loop forever.

Next to the script is a terminal window titled 'Python 3.6.5 Shell'. It shows the output of the script: 'Publish the data' followed by a list of numbers (13, 3, 25, 19, 2, 7, 9) representing the received data.

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

```
Python 3.6.5 Shell
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
Publish the data
```

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	
Data	{"Data":{"temperature":36.4,"humidity":46.5}}	json	a few seconds ago	
Data	{"Data":{"temperature":36.4,"humidity":46.5}}	json	19 minutes ago	
Data	{"Data":{"temperature":36.4,"humidity":46.5}}	json	19 minutes ago	
Data	{"Data":{"temperature":36.4,"humidity":46.5}}	json	19 minutes ago	
Data	{"Data":{"temperature":36.4,"humidity":46.5}}	json	19 minutes ago	

Program:

```
#IBM Watson IOT
Platform #pip install
wiotpsdk import
wiotp.sdk.device import
time import random
myConfig = {"identity":
{
    "orgId": "88653s",
    "typeId": "iot_device",
    "deviceId": "wokwi_us"
},
    "auth": {"token": "1(uiYYO)Nmkr9sk(k")
}
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()
```