

Develop the Python Script

(Publish data to IBM cloud)

Date	04 November 2022
Team ID	PNT2022TMID47980
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 in a terminal. The script, named `publish.py`, imports `paho.mqtt.client`, `time`, and `random`. It defines an `on_publish` function that prints "Publish the data " and a `while True` loop that generates a random number between 1 and 30, publishes it to the `iottopic` with QoS=1, and sleeps for 10 seconds. The terminal output shows the script being restarted and then printing "Publish the data " followed by the numbers 7, 19, 10, and 7.

```
#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 Shell
File Edit Shell Debug Options Window Help
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 informatio
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 in a text editor and its execution in a terminal. The script, named `subscribe.py`, imports `paho.mqtt.client` and defines `on_subscribe` and `on_message` functions. It then creates a `paho.Client` object, sets the callback functions, connects to the `broker.mqttdashboard.com` on port 1883, subscribes to the `iottopic` with QoS=1, and starts the `loop_forever`. The terminal output shows the script receiving and printing "Publish the data " followed by the numbers 13, 3, 25, 19, 2, 7, 9, and 15.

```
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
File Edit Shell Debug Options Window Help
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
```

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A table lists several devices, including 'iot_device_1', 'iot_device_2', 'iot_device_3', and 'wokwi_us'. The 'wokwi_us' device is selected, and its details are shown in a modal window. The modal has tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, displaying a table of events with columns for 'Event', 'Value', 'Format', and 'Last Received'. The events show a stream of data with temperature and humidity values. A status box at the bottom right indicates '5 Simulations running'.

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 wiotp-sdk
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()
```