

# Develop the Python Script

(Publish data to IBM cloud)

|              |  |
|--------------|--|
| Date         | 08 November 2022                                     |
| Team ID      | PNT2022TMID25436                                     |
| Project Name | Industry-specific intelligent fire management system |

## Industry-specific intelligent fire management system



The image shows a Windows desktop environment with two open windows. On the left is a code editor window titled 'Python 3.6.5' containing Python code for publishing data to an MQTT broker. On the right is a terminal window titled 'Python 3.6.5' showing the execution of the script and its output.

**Code Editor Content:**

```
#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)
```

**Terminal Window Output:**

```
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 information.
>>>
===== RESTART: E:/IBM/Others/Develop a python script/
publish.py =====
7
Publish the data
19
Publish the data
10
Publish the data
```

The image shows a split-screen view of a Python IDE. On the left, the code editor displays a Python script for an MQTT client using the paho.mqtt.client library. The code defines an on\_subscribe callback function that prints the subscriber ID and granted QoS. It also defines an on\_message callback function that prints the topic, QoS, and payload. The main loop connects to a broker at 'broker.mqttdashboard.com' on port 1883, subscribes to the topic 'iottopic' with QoS 1, and then enters a loop forever.

```
import paho.mqtt.client as paho
def on_subscribe(client,userdata,mid,granted_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()
```

On the right, the terminal window shows the output of the script. It repeatedly prints the message 'Publish the data' followed by a sequence of integers (13, 1, 3, 25, 19, 2, 7, 9) on separate lines, indicating successful publishes to the topic.

```
Publish the data
13
Publish the data
1
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
```

SIM - Simulation 01 - Windows

ID: (select org)

Browse Action Device Types Interfaces Add Device

DEVICE ID: abc Disconnected DEVICE TYPE: 123 Device LAST PUBLISH: Nov 3, 2022 12:13 PM

Identity Device Information Recent Events State Logs X

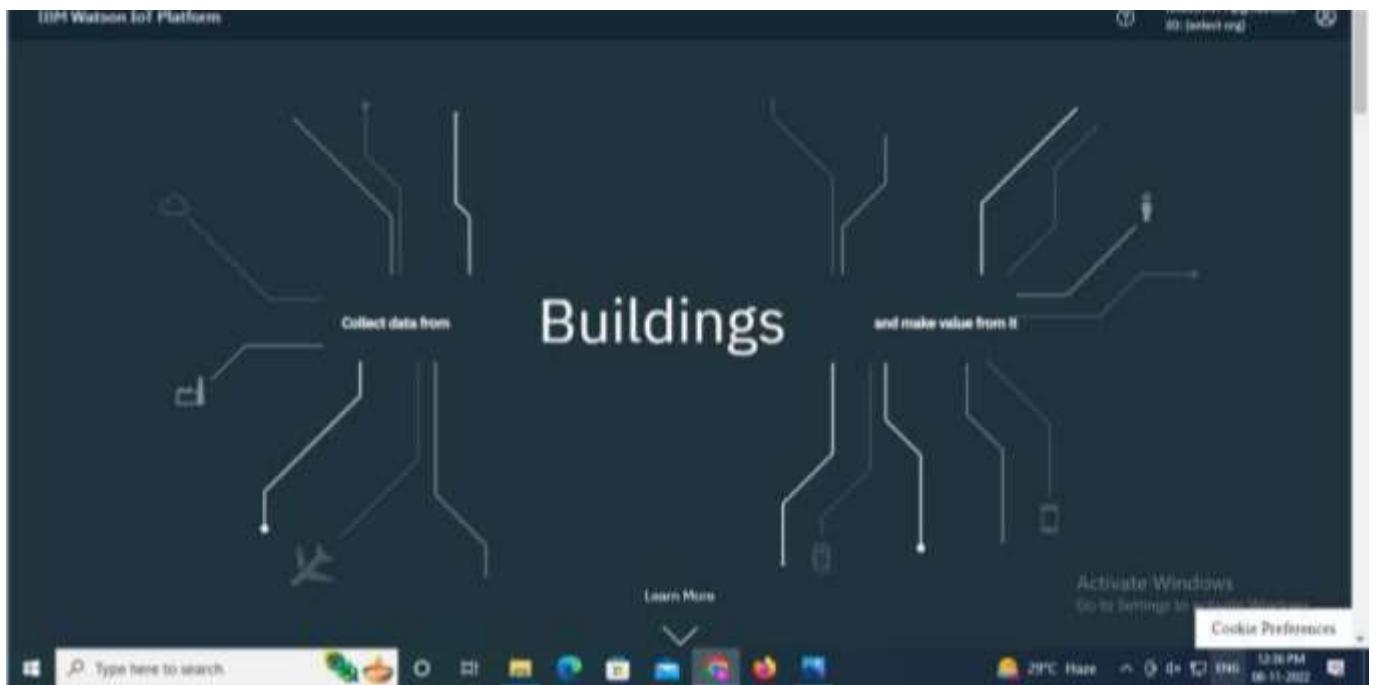
The recent events listed show the live stream of data that is coming and going from this device.

| 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

Type here to search

IBM Screen Node What's note BMC ENG 00:10 04-11-2022



## 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()
```