

Develop the Python Script

(Publish data to IBM cloud)

Team ID	PNT2022TMID10108
Project Name	Industry-specific intelligent fire management system

**Industry-specific intelligent fire management
system**

The image shows a screenshot of a Python script editor and a terminal window. The script, named 'publish.py', is designed to publish data to an MQTT broker. It includes comments and imports for the paho-mqtt library, time, and random modules. The script defines a function 'on_publish' that prints 'Publish the data' and then connects to the broker 'broker.Mqttddashboard.com' on port 1883. It enters a loop where it generates a random number between 1 and 30, publishes it to the topic 'iottopic', and sleeps for 10 seconds. The terminal window shows the output of the script, displaying the message 'Publish the data' three times at different intervals.

```
#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.Mqttddashboard.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 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 image shows a Python IDE window titled 'subscribe.py - E:\BM\Others\Developing a python script\subscribe.py (3.8.3)'. The code defines a MQTT client that subscribes to a topic and prints the received data. The output window shows the following messages:

```
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()
```

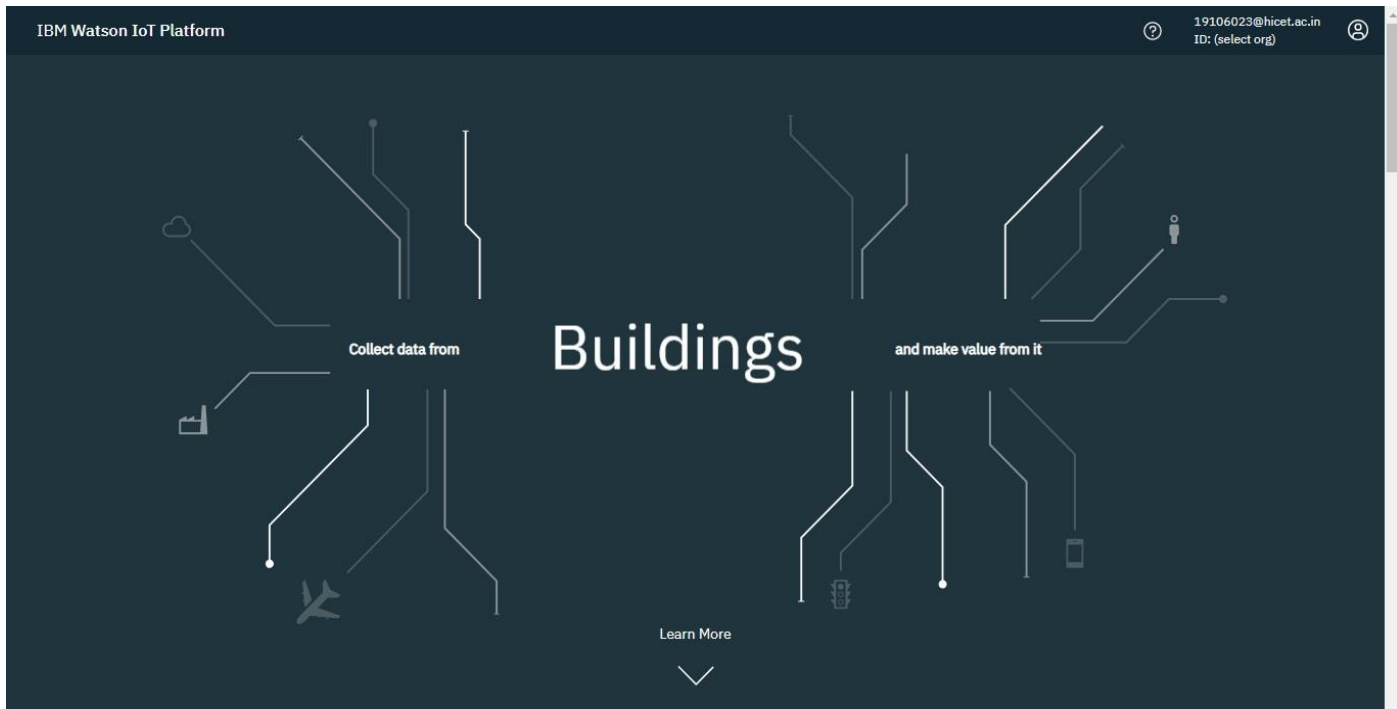
Python 3.8.3 Shell

```
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 image shows a dashboard with tabs: Browse, Action, Device Types, Interfaces. A button 'Add Device' is in the top right. The 'Recent Events' tab is selected, showing a table of events. Below the table, a status box indicates '1 Simulation running'.

Event	Value	Format	Last Received
event_1	{"Temperature":94,"Humidity":95}	json	a few seconds ago
event_1	{"Temperature":73,"Humidity":43}	json	a few seconds ago
event_1	{"Temperature":72,"Humidity":22}	json	a few seconds ago
event_1	{"Temperature":57,"Humidity":55}	json	a few seconds ago
event_1	{"Temperature":64,"Humidity":53}	json	a few seconds ago

1 Simulation running



Powerful web dashboard

Cookie Preferences

Program :

```
#IBM Watson IOT Platform
#pipinstall 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()
```