

Develop the Python Script

(Publish data to IBM cloud)

Date	17 November 2022
Team ID	PNT2022TMID15675
Project Name	Industry-specific intelligent fire management system
Maximum Marks	4 Marks

Industry-specific intelligent fire management system



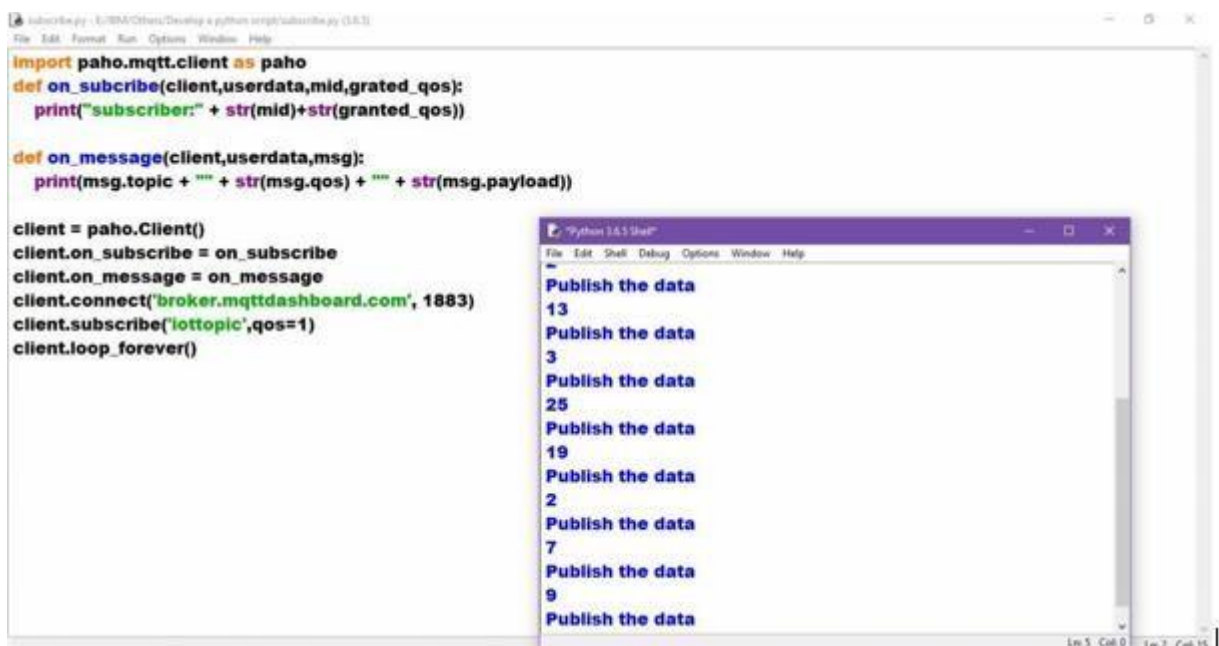
The screenshot shows a Python IDE with a file named 'publish.py'. The code defines a function 'on_publish' that prints 'Publish the data' and then publishes random data to the 'iottopic' on the 'broker.mqttdashboard.com' MQTT broker. The script runs in a loop, publishing data every 10 seconds. A terminal window titled 'Python 3.6.5 Shell' shows the output of the script, displaying 'Publish the data' followed by the random values 7, 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 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 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 IDE with a file named 'subscribe.py'. The code defines functions 'on_subscribe' and 'on_message' to receive data from the 'iottopic' on the 'broker.mqttdashboard.com' MQTT broker. The script connects to the broker and subscribes to the topic. A terminal window titled 'Python 3.6.5 Shell' shows the output of the script, displaying 'Publish the data' followed by the random values 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 displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area shows a device named 'abcd' with status 'Disconnected' and ID '123'. Below this, a tabbed interface shows 'Recent Events' selected. A message states: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this message is a table of recent events:

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

At the bottom of the interface, a status bar indicates '1 Simulation running'. The Windows taskbar at the bottom shows the search bar and several open applications including IBM Watson IoT Platform, Node.js, and various web browsers.

The screenshot shows the 'Buildings' section of the IBM Watson IoT Platform. The background is dark blue with white line art depicting various building and infrastructure elements. The word 'Buildings' is prominently displayed in the center. To the left, the text 'Collect data from' is visible, and to the right, 'and make value from it' is visible. Below the main title, there is a 'Learn More' link. The Windows taskbar at the bottom shows the search bar and several open applications including IBM Watson IoT Platform, Node.js, and various web browsers.

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()

```