

Develop a python script Publish Data to the IBM Cloud

Date	13 November 2022
Team ID	PNT2022TMID33544
Project Name	Project - Signs with smart connectivity for Better road safety
Maximum Marks	4 Marks

Signs with smart connectivity for Better road safety

The screenshot shows a Python script named `publish.py` in a text editor. The script imports `paho.mqtt.client`, `time`, and `random`. It defines an `on_publish` function that prints "Publish the data". The main logic creates a `paho.Client`, connects to `broker.mqttdashboard.com` on port 1883, and enters a `while True` loop. In each iteration, it generates a random integer between 1 and 30, publishes it to the `iottopic` with QoS=1, and sleeps for 10 seconds.

Overlaid on the script is a terminal window titled "Python 3.6.5 Shell" showing the output of the script. It displays the Python version and architecture, followed by a restart message and four lines of output: "Publish the data", "19", "Publish the data", "10", and "Publish the data".

```
#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 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 named `subscribe.py` in a text editor. It imports `paho.mqtt.client` and defines two functions: `on_subscribe`, which prints the subscription status, and `on_message`, which prints the received message topic, QoS, and payload. The main logic creates a `paho.Client`, connects to `broker.mqttdashboard.com` on port 1883, subscribes to the `iottopic` with QoS=1, and calls `client.loop_forever()`.

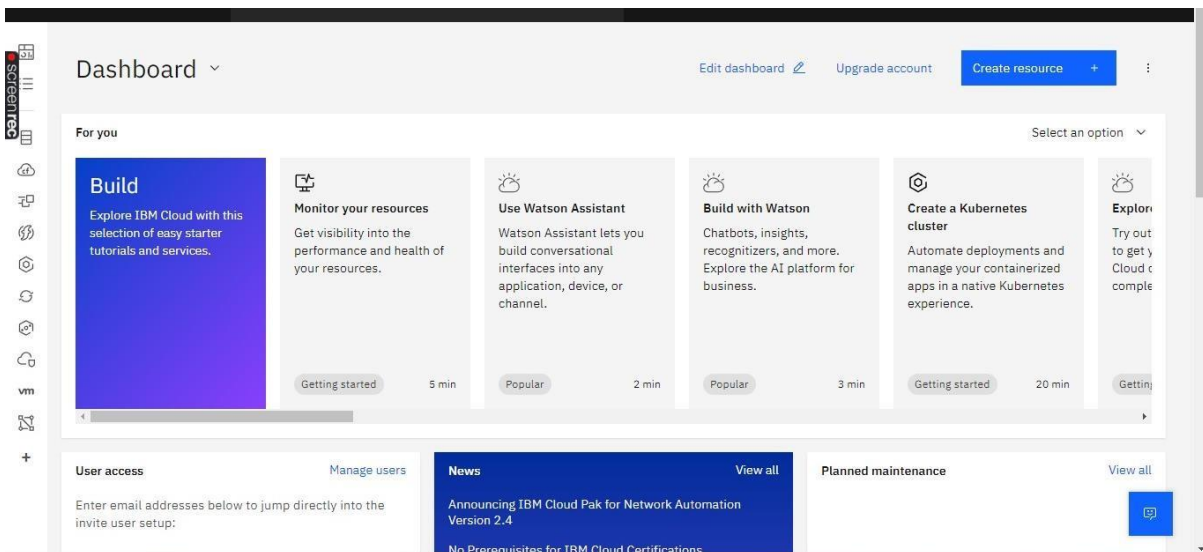
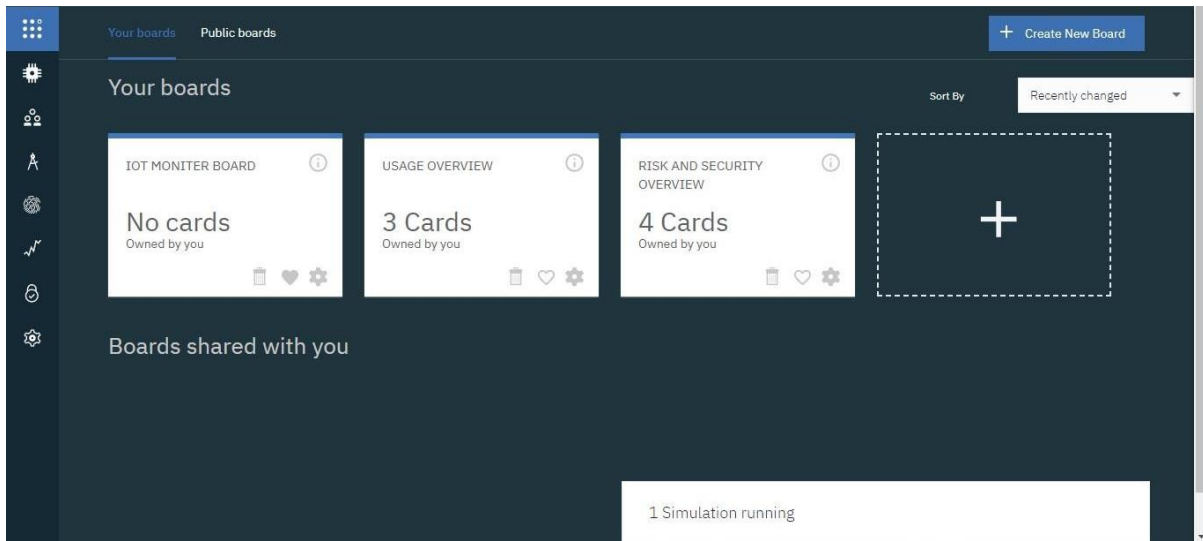
Overlaid on the script is a terminal window titled "Python 3.6.5 Shell" showing the output of the script. It displays the subscription status and then a series of received messages: "Publish the data", "13", "Publish the data", "3", "Publish the data", "25", "Publish the data", "19", "Publish the data", "2", "Publish the data", "7", "Publish the data", "9", and "Publish the data".

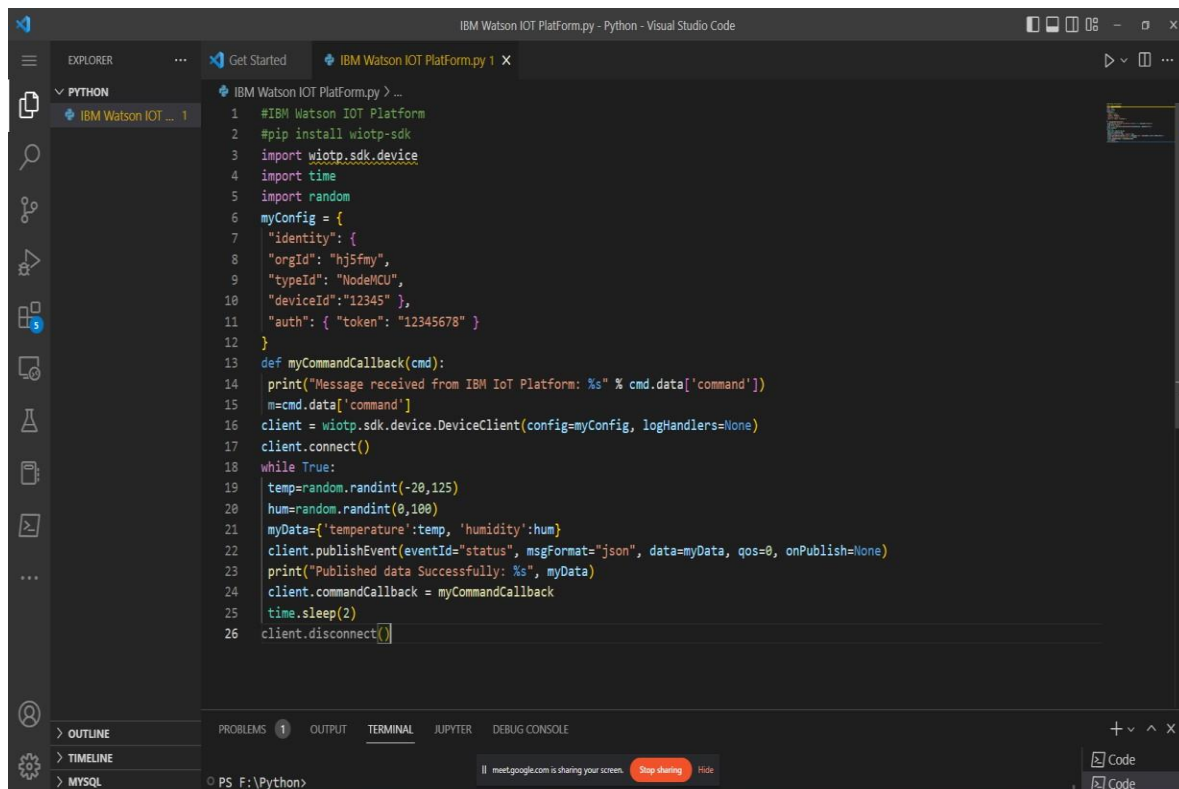
```
import paho.mqtt.client as paho
def on_subscribe(client,userdata,mid,grated_qos):
    print("subscriber:" + str(mid)+str(granted_qos))

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
```





```
1 #IBM Watson IOT Platform
2 #pip install wiotp-sdk
3 import wiotp.sdk.device
4 import time
5 import random
6 myConfig = {
7     "identity": {
8         "orgId": "hj5fmy",
9         "typeId": "NodeMCU",
10        "deviceId": "12345" },
11    "auth": { "token": "12345678" }
12 }
13 def myCommandCallback(cmd):
14     print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
15     m=cmd.data['command']
16     client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
17     client.connect()
18     while True:
19         temp=random.randint(-20,125)
20         hum=random.randint(0,100)
21         myData={'temperature':temp, 'humidity':hum}
22         client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
23         print("Published data Successfully: %s", myData)
24         client.commandCallback = myCommandCallback
25         time.sleep(2)
26     client.disconnect()
```

CODE:

#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()
```