

TEAM ID	PNT2022TMID01883
PROJECT NAME	Smart Waste Management System for Metropolitan Cities

### Publish Data to the IBM Cloud

#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()
```



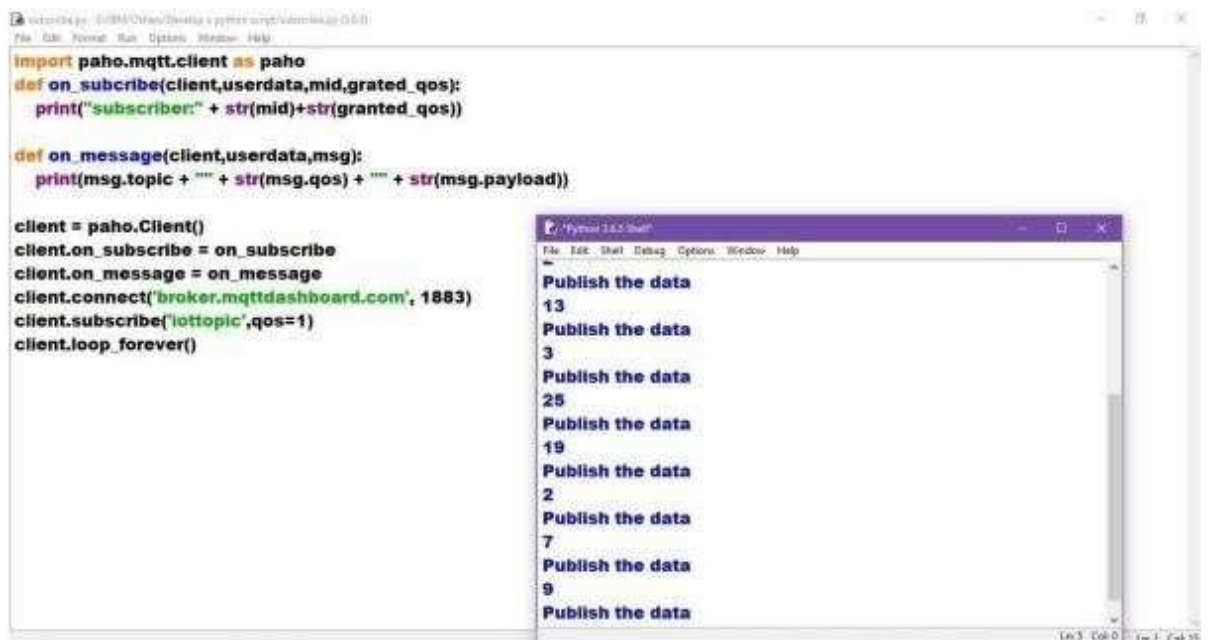
The screenshot shows a Python script in a text editor and its execution in a Python 3.6.5 Shell. The script, titled 'publish.py', imports paho.mqtt.client as paho, time, and random. It defines a function on\_publish that prints 'Publish the data'. The main code creates a paho.Client, sets on\_publish as the callback, connects to broker.mqttdashboard.com on port 1883, starts the loop, and enters a while True loop where it generates a random number between 1 and 30, publishes it to the 'iottopic' with qos=1, prints the value, and sleeps for 10 seconds. The shell output shows the script being restarted, followed by three lines of 'Publish the data' with values 7, 19, 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
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 Python 3.6.5 Shell. The script, titled 'subscriber.py', imports paho.mqtt.client as paho. It defines two functions: on\_subscribe, which prints 'subscriber:' followed by mid and granted\_qos, and on\_message, which prints the message topic, qos, and payload. The main code creates a paho.Client, sets both on\_subscribe and on\_message as callbacks, connects to broker.mqttdashboard.com on port 1883, subscribes to 'iottopic' with qos=1, and calls loop\_forever(). The shell output shows the subscription confirmation and a series of 'Publish the data' messages with various values.

```
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
```

IBM Watson IoT Platform

101 hydraform

Add Device

Browse Action Device Types Interfaces

DEVICE LIST

DEVICE TYPE LAST LOG LAST PING

abcd Disconnected 123 Device Nov 4, 2022 11:51 AM

Identity Device Information Recent Events State Logs

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":75}	json	a few seconds ago
event_1	{"randomNumber":5}	json	a few seconds ago
event_1	{"randomNumber":33}	json	a few seconds ago
event_1	{"randomNumber":56}	json	a few seconds ago
event_1	{"randomNumber":67}	json	a few seconds ago

1 Simulation running

IBM Watson IoT Platform

Simulator Connected to Watson IoT Platform (select org)

Collect data from

# Cars

and make value from it

Learn More