

TEAM ID	PNT2022TMID04297
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 IDE with two windows. The main window displays a Python script for a publisher. The script imports `paho.mqtt.client` as `paho`, `time`, and `random`. It defines a function `on_publish` that prints "Publish the data". The script then creates a `paho.Client` object, sets `on_publish` as the callback, connects to `broker.mqttdashboard.com` on port 1883, and starts the loop. A `while True` loop generates random integers between 1 and 30, publishes them to the topic `iottopic` with `qos=1`, prints the value, and sleeps for 10 seconds.

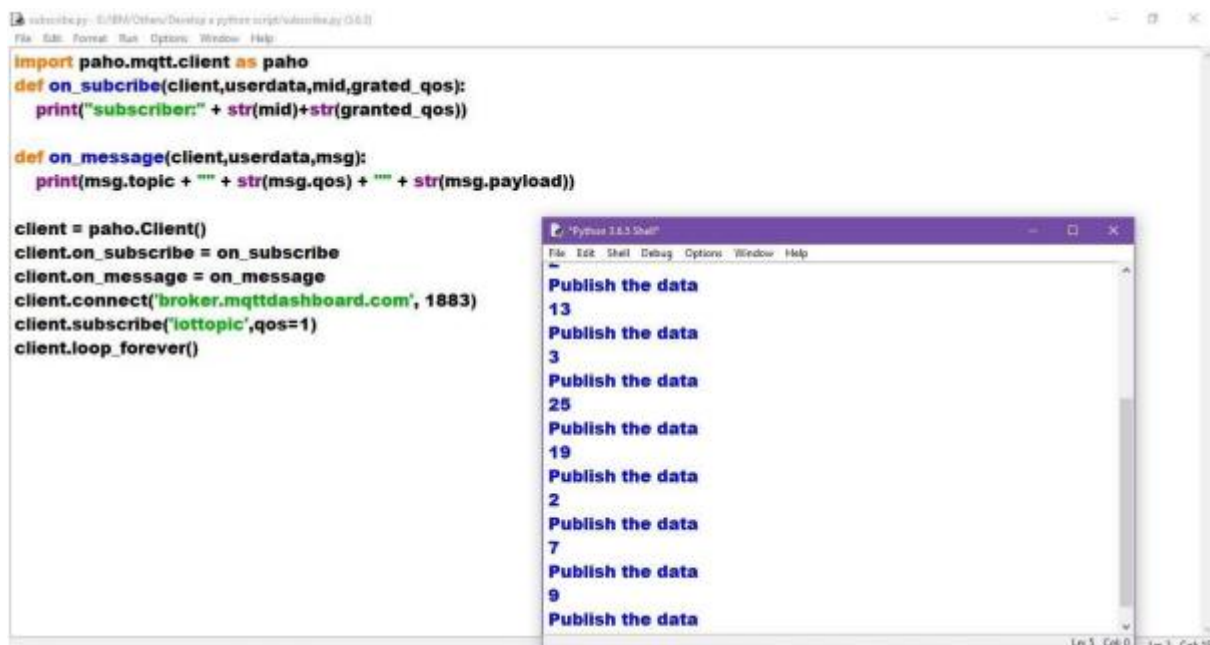
The output window shows the Python 3.6.5 shell output, including the version information and the restart command. The output of the script shows three "Publish the data" messages with values 7, 19, and 10.

```
#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 (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 IDE with two windows. The main window displays a Python script for a subscriber. The script imports `paho.mqtt.client` as `paho`. It defines a function `on_subscribe` that prints the subscriber ID and granted QoS, and a function `on_message` that prints the topic, QoS, and payload. The script then creates a `paho.Client` object, sets the callbacks, connects to `broker.mqttdashboard.com` on port 1883, subscribes to the topic `iottopic` with `qos=1`, and starts the loop.

The output window shows the Python 3.6.5 shell output, including the version information and the restart command. The output of the script shows three "Publish the data" messages with values 13, 3, and 25, followed by three "Publish the data" messages with values 19, 2, and 7, and finally three "Publish the data" messages with values 9, 7, and 9.

```
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

Navigation: Browse | Action | Device Types | Interfaces

Buttons: Add Device

Device ID	Status	Device Type	Last Seen	Last Update
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

Navigation: Browse | Action | Device Types | Interfaces

Buttons: Add Device

# Cars

Collect data from

and make value from it

Learn More