

TEAM ID	PNT2022TMID31883
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": "6lm27j",

"typeId": "IoTensors",

"deviceId": "12345" },

"auth": { "token": "46495064" } }

}

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()
```



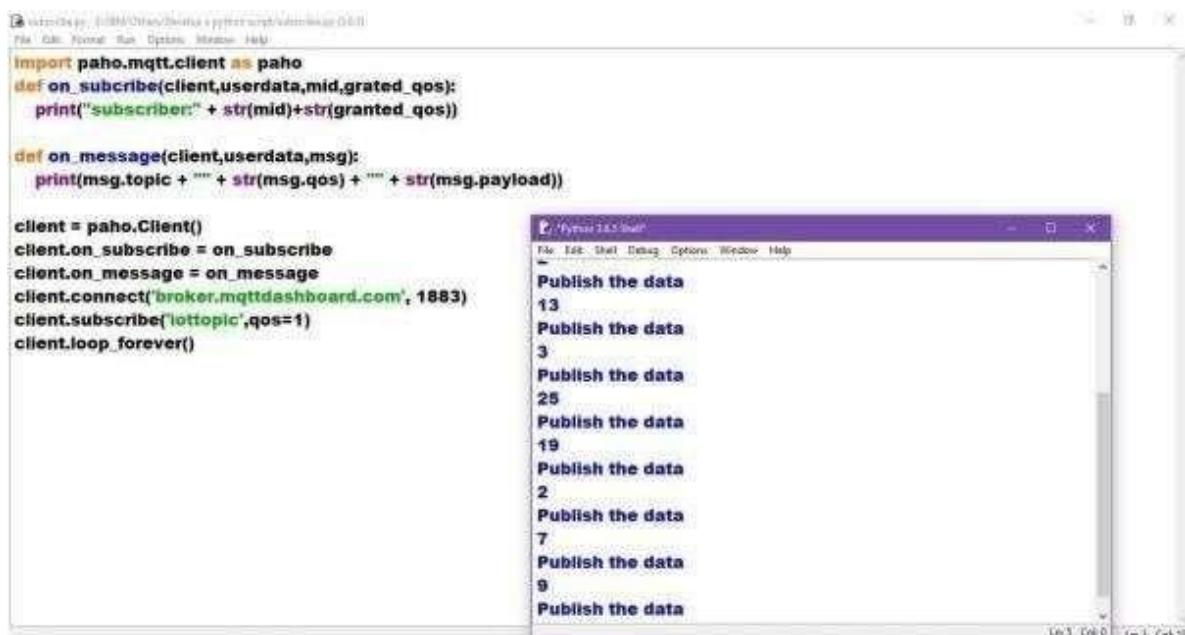
The screenshot shows a Python IDE with a file named 'publish.py'. The code defines an `on_publish` function that prints 'Publish the data' and then publishes a random integer between 1 and 30 to the topic 'iottopic' with a QoS of 1. The script connects to 'broker.mqttdashboard.com' and runs in a loop, publishing data every 10 seconds. A terminal window shows the output of the script, displaying the message 'Publish the data' followed by the values 19, 10, and 7.

```
#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 a file named 'subscriber.py'. The code defines an `on_subscribe` function that prints the subscription details (topic, mid, qos) and an `on_message` function that prints the received message. The script connects to 'broker.mqttdashboard.com' and subscribes to the topic 'iottopic' with a QoS of 1. It then runs in a loop, receiving and printing messages. A terminal window shows the output of the script, displaying the message 'Publish the data' followed by the values 13, 3, 25, 19, 2, 7, 9, and 7.

```
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()
```

```
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

Table Headers: Device ID, Device Name, Device Type, Last Seen, Last Reported

Device Details: abc01, Disconnected, 123, Device, Nov 4, 2022 23:51 AM

Tabs: 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: Action, Device Types, Interfaces

Buttons: Add Device

Section: Cars

Text: Collect data from, and make value from it

Link: Learn More