

**Develop a python script  
Publish Data to the IBM Cloud**

**SMART WASTE MANAGEMENT SYSTEM  
FOR METROPOLITAN CITIES**

**STUDENT NAME : R.HEMALATHA – TEAM LEADER  
M.SANGEETHA – TEAM MEMBER  
P.ABIRAMI -TEAM MEMBER  
S.JENIFA MARY – TEAM MEMBER**

The screenshot shows a Python script in a text editor and its execution in a shell. The script, named `publish.py`, imports `paho.mqtt.client`, `time`, and `random`. It defines a function `on_publish` 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 the MQTT broker `broker.mqttdashboard.com` and starts a loop that continuously publishes data every 10 seconds.

```
#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)
```

The shell output shows the script being restarted and the following data being published:

```
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 script in a text editor and its execution in a shell. The script, named `subscribe.py`, imports `paho.mqtt.client` and defines two functions: `on_subscribe` and `on_message`. `on_subscribe` prints "subscriber:" followed by the mid and granted QoS. `on_message` prints the topic, QoS, and payload. The script connects to the MQTT broker `broker.mqttdashboard.com`, subscribes to the topic `iottopic` with a QoS of 1, and starts a loop that continuously receives and prints messages.

```
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()
```

The shell output shows the script being restarted and the following data being received:

```
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 shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Service Details - IBM Cloud' and 'IBM Watson IoT Platform'. The main header displays the user's email 'saravanasudhansubramanian2002@gmail.com' and ID 'hqbs6m'. The dashboard is divided into sections: 'Browse', 'Action', 'Device Types', and 'Interfaces'. A table lists devices, with 'abcd' selected. The 'Recent Events' tab is active, showing a stream of data events. Below the table, it indicates '1 Simulation running'.

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

The screenshot shows the IBM Watson IoT Platform landing page. The top navigation bar includes 'Service Details - IBM Cloud' and 'IBM Watson IoT Platform'. The main header displays the user's email 'saravanasudhansubrama...' and ID '(select org)'. The landing page features a dark background with a circuit-like pattern. The central text reads 'Cars'. To the left, it says 'Collect data from' with an icon of a car. To the right, it says 'and make value from it' with an icon of a car. A 'Learn More' button is visible at the bottom center. A 'Cookie Preferences' button is located in the bottom right corner.

## 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()

```