

DEVELOP THE PYTHON SCRIPT(PUBLISH DATA TO IBM CLOUD)

IBM ID	IBM-Project-6081-1658823192
Team ID	PNT2022TMID10960
Project Name	Industry-Specific Intelligent Fire Management System



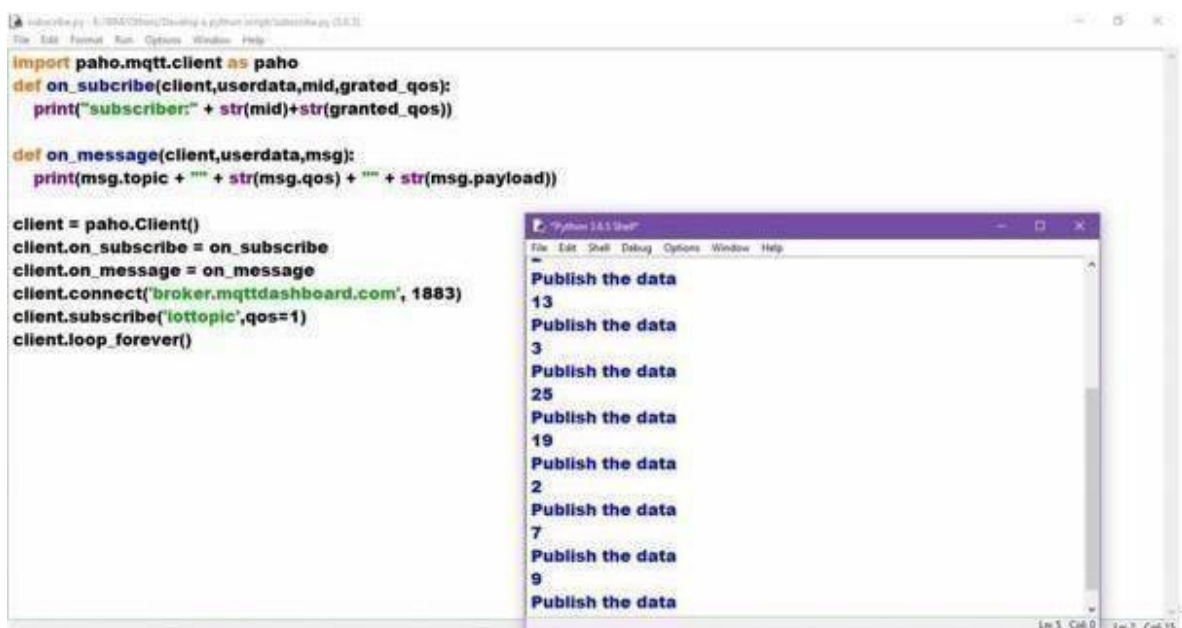
The screenshot shows a Python IDE with a file named 'publish.py'. The code defines a function 'on_publish' that prints 'Publish the data' and then publishes a random number to the 'iottopic' on the 'broker.mqttdashboard.com' MQTT broker. The main code creates a Paho MQTT client, connects to the broker, and enters a loop that calls 'on_publish' every 10 seconds. An inset window titled 'Python 3.6.5 Shell' shows the output of the script, displaying the message 'Publish the data' followed by random numbers: 7, 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 information.
>>>
===== 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 'subscribe.py'. The code defines two functions: 'on_subscribe' which prints the subscriber ID and granted QoS, and 'on_message' which prints the topic, QoS, and payload. The main code creates a Paho MQTT client, connects to the broker, subscribes to the 'iottopic', and enters a loop that calls 'on_message' for every received message. An inset window titled 'Python 3.6.5 Shell' shows the output of the script, displaying the message 'Publish the data' followed by random numbers: 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))

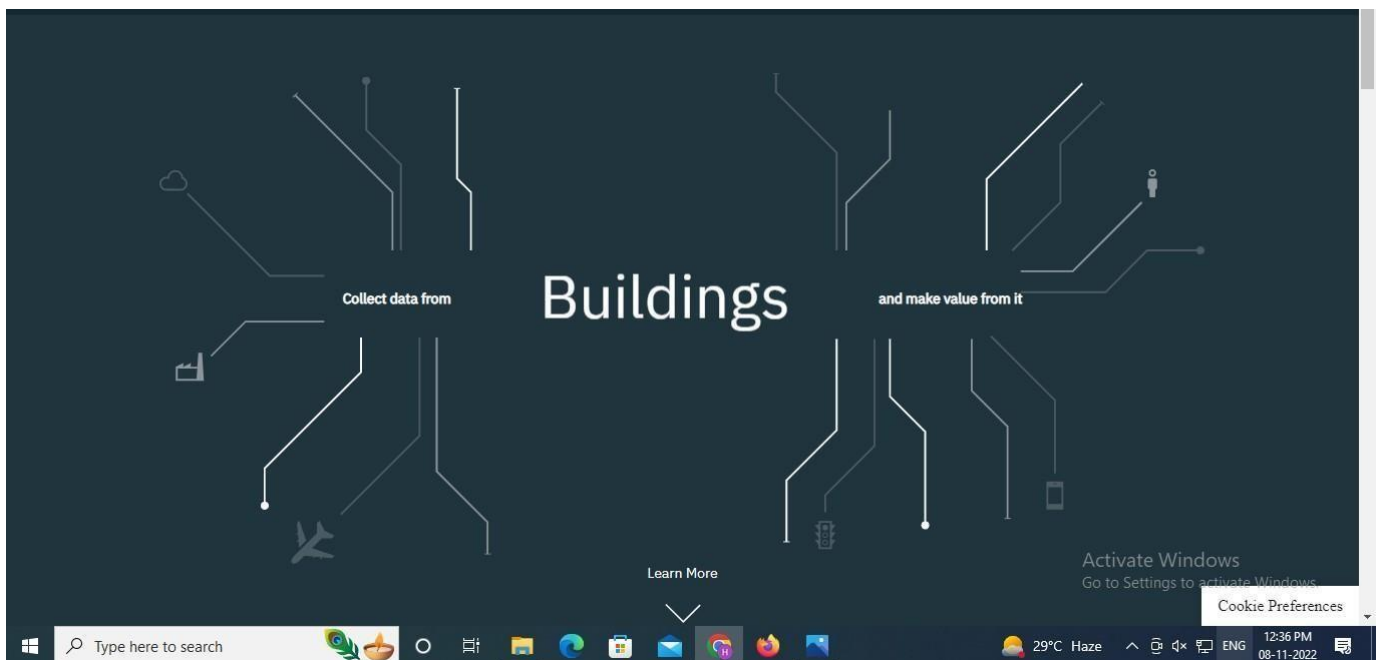
client = paho.Client()
client.on_subscribe = on_subscribe
client.on_message = on_message
client.connect('broker.mqttdashboard.com', 1883)
client.subscribe('iottopic',qos=1)
client.loop_forever()
```

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

The screenshot shows the IBM Cloud IoT Platform console. At the top, there's a navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search icon is on the left, and an 'Add Device' button is on the right. Below this is a table of devices. The first device is 'abcd', with status 'Disconnected', ID '123', type 'Device', and last seen 'Nov 3, 2022 12:13 PM'. A dropdown menu is open for this device, showing tabs: 'Identity', 'Device Information', 'Recent Events' (selected), 'State', and 'Logs'. Under 'Recent Events', there's a message: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this is a table of events:

Event	Value	Format	Last Received
event_1	{"randomNumber":74}	json	a few seconds ago
event_1	{"randomNumber":47}	json	a few seconds ago
event_1	{"randomNumber":45}	json	a minute ago
event_1	{"randomNumber":19}	json	a minute ago
event_1	{"randomNumber":79}	json	a minute ago

At the bottom of the console, a status bar shows '1 Simulation running'.



Program :

#IBM Watson IOT
Platform

```
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
```

```
myConfig = {"identity":
{
    "orgId": "f68qgi",
    "typeId": "NodeMCU",
    "deviceId": "2345"},
    "auth": {"token": "12345678"}}
```

```
def myCommandCallback(cmd): print ("Message received from IBM
IoTPlatform: %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()
```