

DEVELOP THE PYTHON SCRIPT

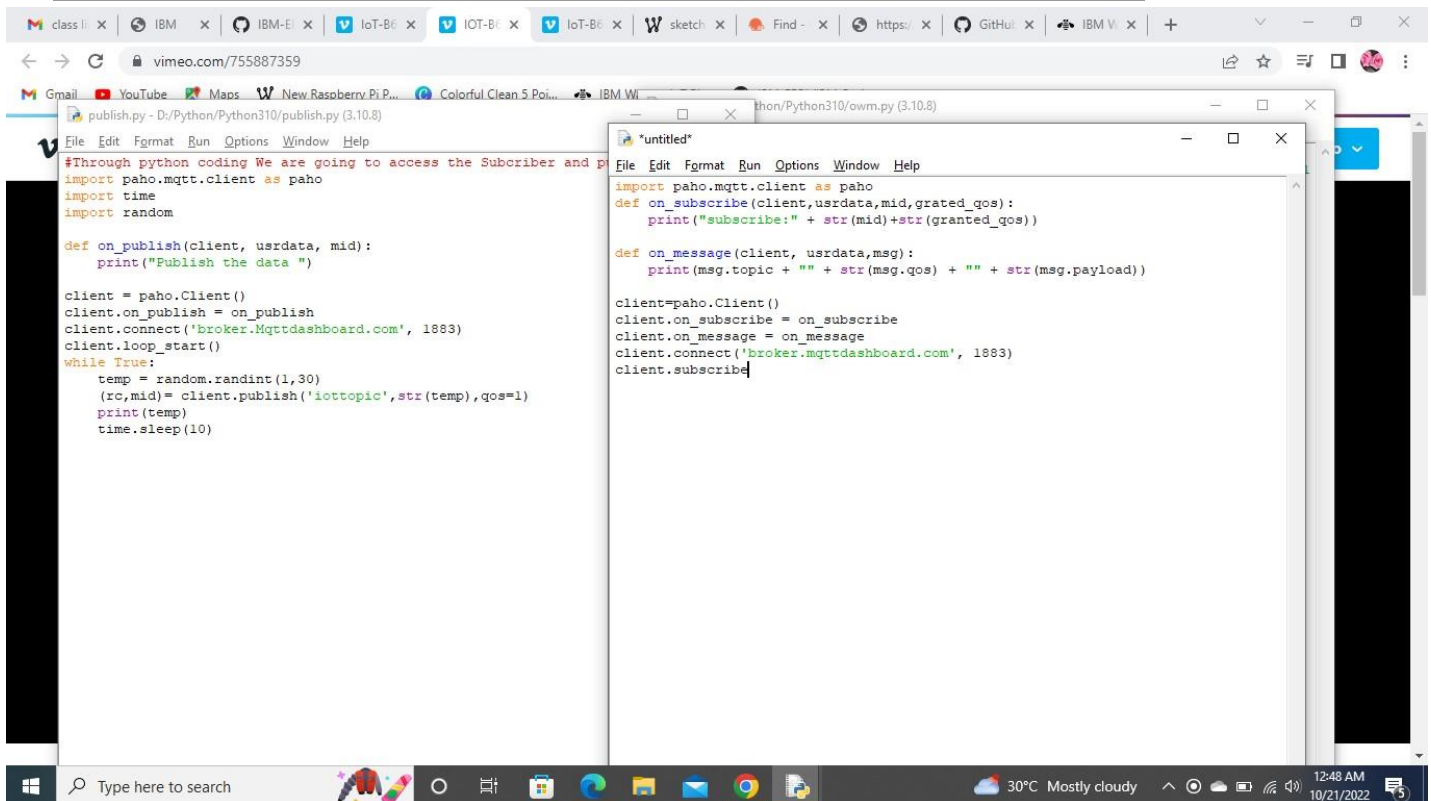
Publish data to the IBM Cloud

Date : 21 October 2022

Team ID : PNT2022TMID48096

Project Name - SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

To make a Publisher and Subscriber in the process of Python and IBM cloud



The screenshot shows a Windows desktop environment. In the background, a video player is open to a Vimeo page. In the foreground, two code editors are visible. The left editor, titled 'publish.py', contains Python code for a publisher. The right editor, titled '*untitled*', contains Python code for a subscriber. Both scripts use the paho-mqtt library to connect to a broker at 'broker.mqttdashboard.com' on port 1883. The publisher script generates random temperature data and publishes it to the 'iottopic' topic. The subscriber script listens for messages on the 'iottopic' topic and prints them.

```
#Through python coding We are going to access the Subscriber and p
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)
    (rc,mid)= client.publish('iottopic',str(temp),qos=1)
    print(temp)
    time.sleep(10)
```

```
import paho.mqtt.client as paho

def on_subscribe(client,userdata,mid,grated_qos):
    print("subscribe:" + 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
```

class | x IBM | x IBM-E | x IoT-B6 | x IoT-B6 | x IoT-B6 | x sketch | x Find | x https:// | x GitHub | x IBM V | x +

vimeo.com/755887359

Gmail | YouTube | Maps | W New Raspberry Pi P... | Colorful Clean 5 Poi... | IBM Watson IoT Pla... | IBM-EPBL/IBM-Proj...

publish.py - D:/Python/Python310/publish.py (3.10.8)

File Edit Format Run Options Window Help

#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)
    (rc,mid)= client.publish('iottopic',str(temp),qos=1)
    print(temp)
    time.sleep(10)
```

Command Prompt - python publish.py

```
C:\Users\USER>python publish.py
15
Publish the data
4
Publish the data
```

28°C 11:00 AM 01-Oct-22

Type here to search

class | x IBM | x IBM-E | x IoT-B6 | x IoT-B6 | x IoT-B6 | x sketch | x Find | x https:// | x GitHub | x IBM V | x +

vimeo.com/755887359

Gmail | YouTube | Maps | W New Raspberry Pi P... | Colorful Clean 5 Poi... | IBM Watson IoT Pla... | IBM-EPBL/IBM-Proj...

vimeo Solutions Features Resources Watch

Command Prompt - python subscribe.py

```
{'coord
p': 30}Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.
9880,
```

Command Prompt - python publish.py

```
C:\Users\USER>python publish.py
15
Publish the data
4
Publish the data
9
Publish the data
1
Publish the data
9
Publish the data
24
Publish the data
16
Publish the data
11
Publish the data
13
Publish the data
26
Publish the data
17
Publish the data
```

subscribe.py - D:/Python/Python310/subscribe.py (3.10.8)

File Edit Format Run Options Window Help

```
import paho.mqtt.client as paho
def on_subscribe(client,userdata,mid,grated_qos):
    print("subscribe:" + str(mid)+str(grated_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()
```

30°C Mostly cloudy 12:54 AM 10/21/2022

IBM Watson IoT Platform

410119106009@smartinternz.com
ID: 7kb3es

[Your boards](#) [Public boards](#) [+ Create New Board](#)

IOT MONITOR BOARD

No cards

Owned by you

USAGE OVERVIEW

3 Cards

Owned by you

RISK AND SECURITY OVERVIEW

4 Cards

Owned by you

+

Boards shared with you

1 Simulation running

Type here to search

31°C Cloudy 11:14 PM 10/15/2022

IBM Watson IoT Platform

410119106009@smartinternz.com
ID: 7kb3es

[Iot monitor board](#) [+ Add New Card](#) [Settings](#)

Line chart

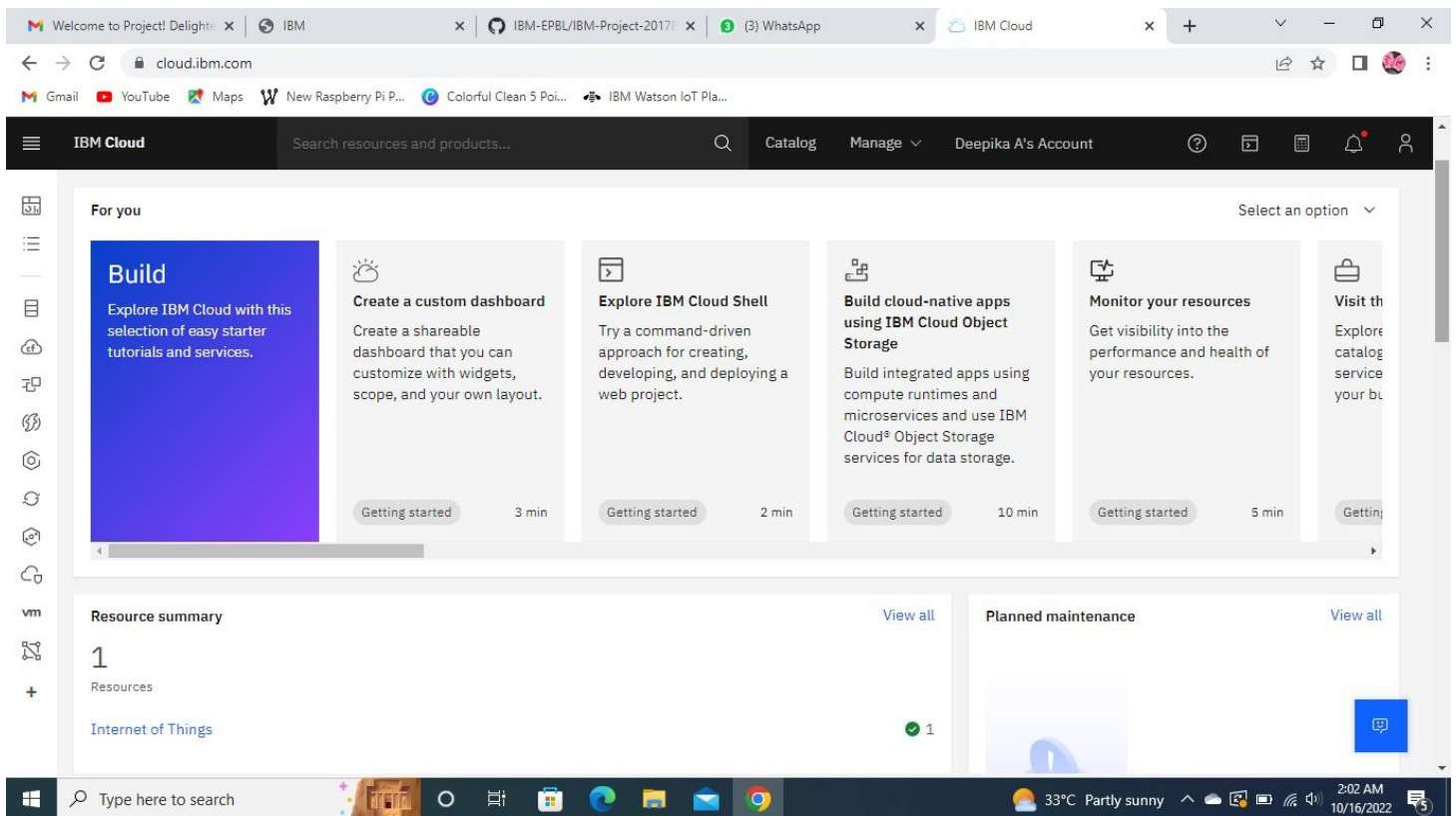
5 minutes [now](#)

randomNumber sampleObject.xcord sampleObject.ycord

1 Simulation running

Type here to search

31°C Cloudy 11:21 PM 10/15/2022



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()

```