

## DEVELOP A PYTHON SCRIPT

### Publish Data to the IBM Cloud

Date	31 october 2022
Team Id	PNT2022TMID12798
Project Name	Project - Signs with smart connectivity for Better road safety

### Signs with smart connectivity for Better road safety

#### Python code to access 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)

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 screenshot shows a Windows desktop environment. On the left, a text editor window titled 'subscribe.py - C:/Python/Python37/subscribe.py (3.7.4)' contains the following Python code:

```

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)

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()

```

On the right, a 'Python 3.7.4 Shell' window shows the output of the script. It displays a restart message and a series of 'Publish the data' messages with increasing line numbers (30, 22, 27, 27, 7, 3, 3).

The Windows taskbar at the bottom shows the search bar, task view button, and several pinned application icons including File Explorer, Edge, and various utility tools. The system tray on the right indicates the date and time as 09-11-2022, 02:34.

## PROGRAM:

#IBM Watson IOT Platform

#pip install wiotp-sdk

import wiotp.sdk.device

import time

import random

myConfig = {

"identity": {

"orgId": "gsqz5f",

"typeId": "NANDY",

"deviceId": "12345" },

"auth": { "token": "9876543210" }

}

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 script running in a shell window. The script defines a configuration for the IBM IoT Platform, connects a client, and enters a loop where it publishes data (temperature and humidity) to the platform. The output shows a series of 'Published data Successfully' messages with varying temperature and humidity values.

```

publish.py - C:/Python/Python37/publish.py (3.7.4)
File Edit Format Run Options Window Help

import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "qsqs5f",
        "typeId": "NANDY",
        "deviceId": "12345" },
    "auth": { "token": "9876543210" }
}
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()

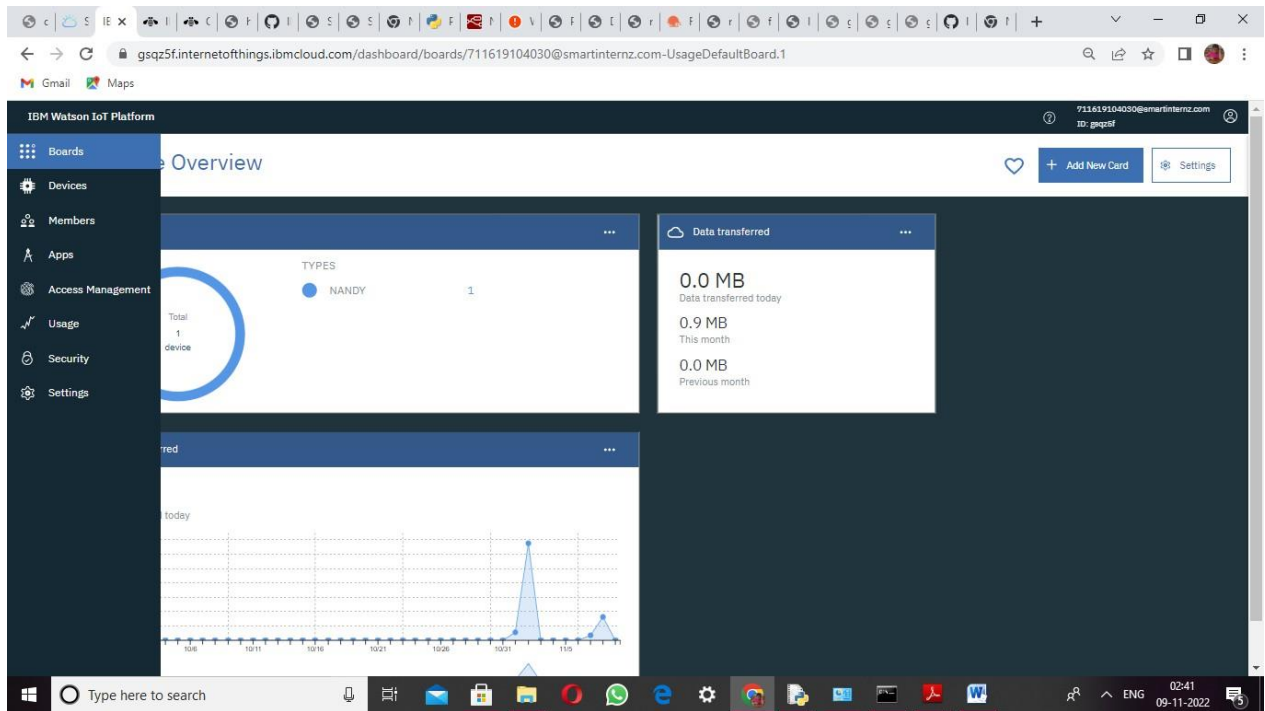
```

```

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Published data Successfully: %s ('temperature': 50, 'humidity': 45)
Published data Successfully: %s ('temperature': 8, 'humidity': 28)
Published data Successfully: %s ('temperature': 46, 'humidity': 1)
Published data Successfully: %s ('temperature': 78, 'humidity': 49)
Published data Successfully: %s ('temperature': 81, 'humidity': 41)
Published data Successfully: %s ('temperature': 73, 'humidity': 80)
Published data Successfully: %s ('temperature': 76, 'humidity': 34)
Published data Successfully: %s ('temperature': 2, 'humidity': 81)
Published data Successfully: %s ('temperature': 33, 'humidity': 32)
Published data Successfully: %s ('temperature': 18, 'humidity': 76)
Published data Successfully: %s ('temperature': 68, 'humidity': 75)
Published data Successfully: %s ('temperature': 85, 'humidity': 39)
Published data Successfully: %s ('temperature': -10, 'humidity': 96)
Published data Successfully: %s ('temperature': 112, 'humidity': 23)
Published data Successfully: %s ('temperature': 63, 'humidity': 52)
Published data Successfully: %s ('temperature': -18, 'humidity': 36)
Published data Successfully: %s ('temperature': 103, 'humidity': 70)
Published data Successfully: %s ('temperature': 39, 'humidity': 12)
Published data Successfully: %s ('temperature': 6, 'humidity': 51)
Published data Successfully: %s ('temperature': 0, 'humidity': 71)
Published data Successfully: %s ('temperature': 76, 'humidity': 24)
Published data Successfully: %s ('temperature': -16, 'humidity': 73)
Published data Successfully: %s ('temperature': -8, 'humidity': 35)
Published data Successfully: %s ('temperature': 60, 'humidity': 49)
Published data Successfully: %s ('temperature': 79, 'humidity': 97)
Published data Successfully: %s ('temperature': 93, 'humidity': 72)
Published data Successfully: %s ('temperature': -13, 'humidity': 72)
Published data Successfully: %s ('temperature': 67, 'humidity': 90)
Published data Successfully: %s ('temperature': 108, 'humidity': 83)
Published data Successfully: %s ('temperature': 71, 'humidity': 32)
Published data Successfully: %s ('temperature': 47, 'humidity': 75)
Published data Successfully: %s ('temperature': 65, 'humidity': 16)
Published data Successfully: %s ('temperature': 10, 'humidity': 83)
Published data Successfully: %s ('temperature': 24, 'humidity': 76)
Published data Successfully: %s ('temperature': 109, 'humidity': 31)
Published data Successfully: %s ('temperature': 15, 'humidity': 24)
Published data Successfully: %s ('temperature': 0, 'humidity': 34)
Published data Successfully: %s ('temperature': 44, 'humidity': 87)
Published data Successfully: %s ('temperature': 99, 'humidity': 94)
Ln 5

```

Publish the data to the ibm cloud:



The screenshot shows the IBM Watson IoT Platform dashboard, specifically the 'Devices' page. The page has tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A table lists the devices, and a detailed view for device 12345 is shown below.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By	Device Class
12345	Connected	NANDY	Device	Nov 3, 2022 12:12 AM		711619104030@smartinternz-com	

Below the table, a detailed view for device 12345 is shown:

Identity	Device Information	Recent Events	State	Logs
Device ID	12345			
Device Type	NANDY			
Date Added	Nov 3, 2022 12:12 AM			
Added By	711619104030@smartinternz-com			
Connection Status	Connected			
	Connection Time: Nov 9, 2022 2:40 AM			
	Client Address: 106.222.125.219 SecureToken			

At the bottom, there is a pagination bar showing 'Items per page 50' and '1-1 of 1 item'.



