

DEVELOP A PYTHON SCRIPT

Publish Data to the IBM Cloud

Date	31 october 2022
Team Id	PNT2022TMID42771
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:

```

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Python/Python37/subscribe.py =====
30
Publish the data
22
Publish the data
27
Publish the data
27
Publish the data
7
Publish the data
3
Publish the data

```

The Windows taskbar at the bottom shows the Start button, a search bar, and several 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 Windows desktop environment. On the left, a text editor window titled 'publish.py - C:/Python/Python37/publish.py (3.7.4)' contains the Python code. On the right, a terminal window titled '*Python 3.7.4 Shell*' displays the output of the script. The output consists of multiple lines of 'Published data Successfully' messages, each followed by a JSON object containing 'temperature' and 'humidity' values. The terminal window also shows a status bar at the bottom with 'Ln: 5' and 'Col: 5'.

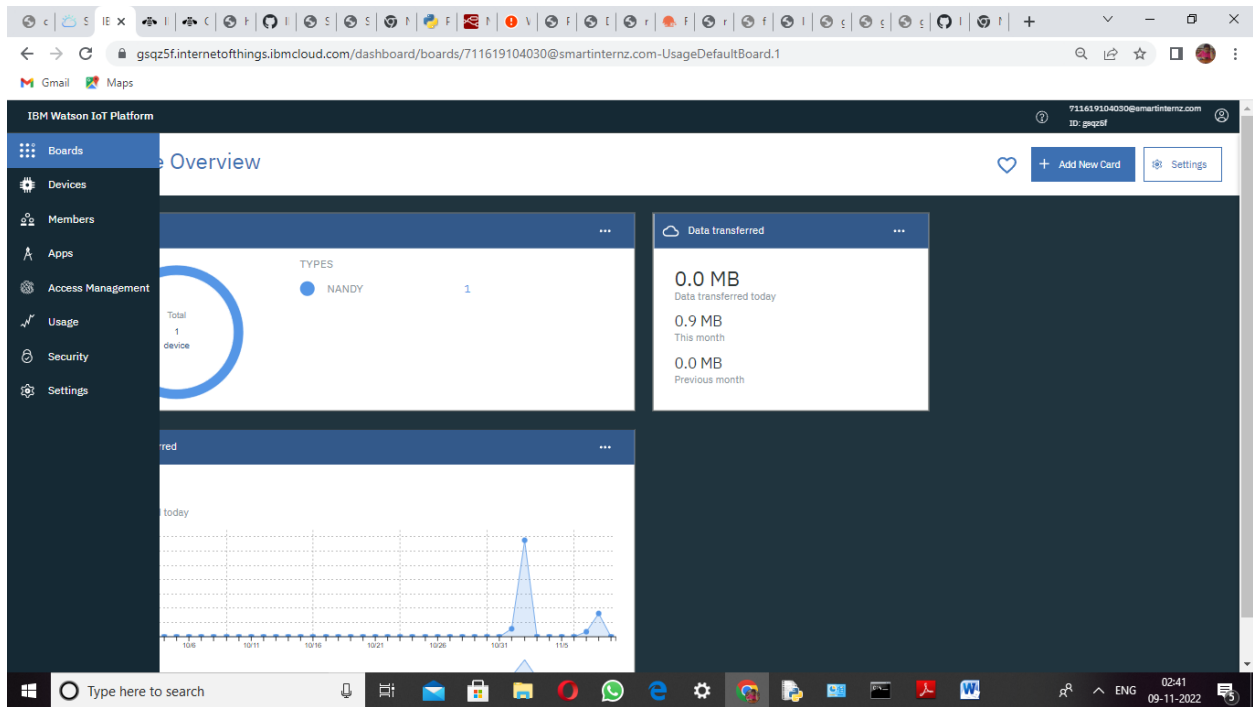
```

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": "gsqz5f",
        "typeId": "NANDY",
        "deviceId": "12345" },
    "auth": { "token": "9076543210" }
}
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': 79)
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': 38, '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 Col: 5

```

Publish the data to the ibm cloud:



The screenshot shows the IBM Watson IoT Platform Devices page. The left sidebar contains navigation links: Browse, Action, Device Types, and Interfaces. The main content area displays a table of devices. The table has columns: Device ID, Status, Device Type, Class ID, Date Added, Descriptive Location, Added By, and Device Class. The table shows one device with ID 12345, Status Connected, Device Type NANDY, Class ID Device, Date Added Nov 3, 2022 12:12 AM, Descriptive Location, Added By 711619104030@smartinternz.com, and Device Class. Below the table, there is a section for 'Device Information' with details: Device ID 12345, Device Type NANDY, Date Added Nov 3, 2022 12:12 AM, Added By 711619104030@smartinternz.com, and Connection Status Connected. The connection status details include Connection Time: Nov 9, 2022 2:40 AM and Client Address: 106.222.128.219 SecureToken.

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	

Device Information

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	106.222.128.219 SecureToken

