

DEVELOP THE PYTHON SCRIPT (PUBLISH DATA TO IBM CLOUD)

Team ID	PNT2022TMID30584
Project Name	Industry-Specific Intelligent Fire Management System



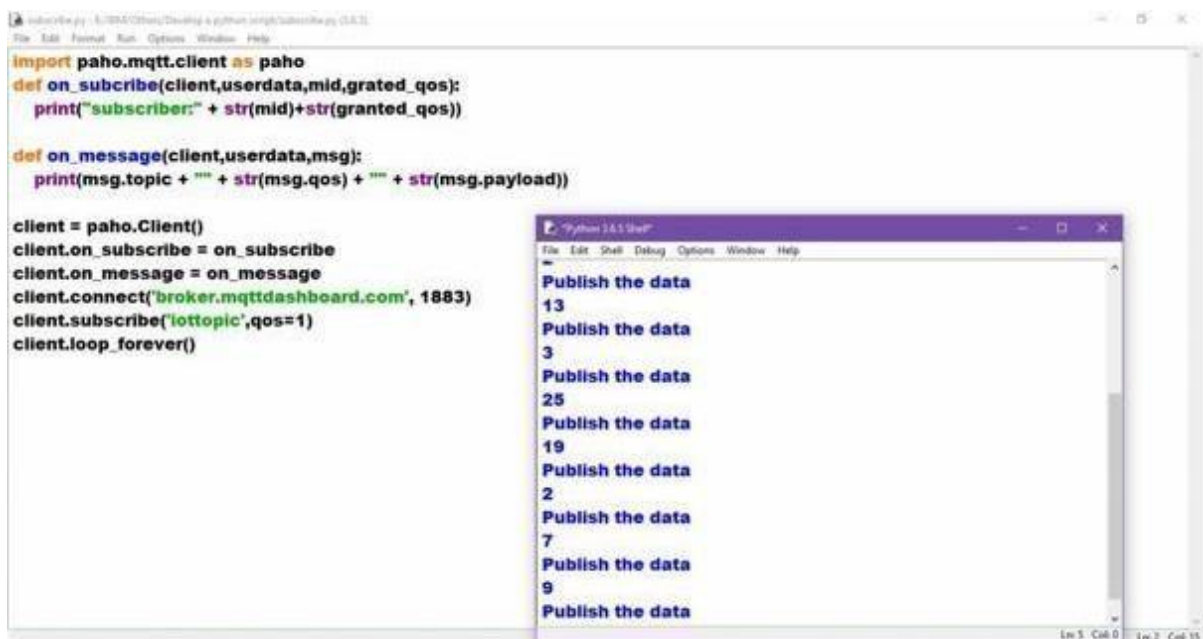
```
#Through python coding we are going to access the subscriber
import paho.mqtt.client as paho
import time
import random

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 Shell
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 informatio
n.
>>>
===== RESTART: E:/IBM/Others/Develop a python script/
publish.py =====
7
Publish the data
19
Publish the data
10
Publish the data
```



```
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 Cisco DevNet sandbox interface. At the top, there are tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue button labeled 'Add Device' with a plus icon is in the top right. Below the tabs, a table lists devices. The first device is 'abcd', status 'Disconnected', ID '123', type 'Device', and added on 'Nov 3, 2022 12:13 PM'. Below the device list, the 'Recent Events' tab is selected, showing a table of events. The table has columns: Event, Value, Format, and Last Received. The events are 'event_1' with values like '{\"randomNumber\":74}', '{\"randomNumber\":47}', '{\"randomNumber\":45}', '{\"randomNumber\":19}', and '{\"randomNumber\":79}', all in 'json' format, with timestamps like 'a few seconds ago' and 'a minute ago'. A status bar at the bottom indicates '1 Simulation running'.

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

1 Simulation running



Program :

```
#IBM Watson IOT
Platform

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

myConfig = {"identity":
{
    "orgId": "sms611",
    "typeId": "3114",
    "deviceId": "14"},
    "auth": {"token": "753998223"}
}

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("PublisheddataSuccessfully:%s",myData)
client.commandCallback = myCommandCallback
time.sleep(2)
client.disconnect()
```