

PYTHON SCRIPT TO PUBLISH DATA TO IBM CLOUD

Date	13 NOVEMBER 2022
Team ID	PNT2022TMID43114
Project Name	SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES
Maximum marks	8 Marks

PYTHON CODE:

```
#IBM Watson IOT Platform
```

```
#pip install wiotp-sdk
```

```
import wiotp.sdk.device
```

```
import time
```

```
import random
```

```
myConfig = {
```

```
    "identity": {
```

```
        "orgId": "80bbqy",
```

```
        "typeId": "cse2019",
```

```
        "deviceId": "ganesh1601"
```

```
    },
```

```
    "auth": {
```

```
        "token": "!8-vkyUJA+QxM-L+uf"
```

```
    }
```

```
}
```

```
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 CODE:

```
legacy-C:\Users\ECOT\Desktop\legacy\legacy3.1.4
File Edit Format Run Options Window Help

#IBM Watson IoT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "Alliifu",
        "typeId": "blindeddevicetype",
        "deviceId": "44327device"
    },
    "auth": {
        "token": "3nj82u8H3de8n7ou"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandler=None)
client.connect()

while True:
    temp=random.randint(-20,35)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="datacmd", msgFormat="json", data=myData, qos=0, onSuccess=None)
    print("Published data Successfully %s" % myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

PYTHON OUTPUT:

```
*Python 3.7.4 Shell*
File Edit Shell Debug Options Window Help
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:/Users/ELCOT/OneDrive/Desktop/loga.py =====
2022-11-12 16:03:37,102 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:x3lifo:b1lm3edevicetype:43127device
Published data Successfully: %s ('temperature': 54, 'humidity': 94)
Published data Successfully: %s ('temperature': 80, 'humidity': 27)
Published data Successfully: %s ('temperature': 90, 'humidity': 23)
Published data Successfully: %s ('temperature': 36, 'humidity': 6)
Published data Successfully: %s ('temperature': 121, 'humidity': 24)
Published data Successfully: %s ('temperature': 1, 'humidity': 16)
Published data Successfully: %s ('temperature': 20, 'humidity': 84)
|
```

Ln: 12 Col: 0

```
logpy - C:\Users\KLOOT\OneDrive\Desktop\logpy 0.7.0
File Edit Format Run Options Window Help

#100 Watson IoT Platform
#pip install wattp-edk
import wattp.edk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "e911fe",
        "appId": "h1n0edevicetype",
        "deviceId": "43327device"
    },
    "auth": {
        "token": "0uq92u0N2u0F0u7e0t"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform %s" % cmd.data['command'])
    myCmd.data['command']

client = wattp.edk.device.TwiceClient(config=myConfig, logHandler=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,
        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

Python 3.7.4 (tags/v3.7.4:00059132e, Jul 8 2019, 20:54:10) [AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\KLOOT\OneDrive\Desktop\logpy =====
2020-11-12 15:57:00.614 wattp.edk.device.client.DeviceClient INFO Connects
d successfully: d1431176c31b0edevicetype43327device
Published data Successfully: %s ('temperature': 151, 'humidity': 23)
Published data Successfully: %s ('temperature': -4, 'humidity': 44)
Published data Successfully: %s ('temperature': 29, 'humidity': 50)
Published data Successfully: %s ('temperature': 1, 'humidity': 10)
Published data Successfully: %s ('temperature': 87, 'humidity': 41)
Published data Successfully: %s ('temperature': 11, 'humidity': 30)
Published data Successfully: %s ('temperature': 87, 'humidity': 30)
Published data Successfully: %s ('temperature': 104, 'humidity': 53)
Published data Successfully: %s ('temperature': 10, 'humidity': 89)
Published data Successfully: %s ('temperature': 77, 'humidity': 60)
Published data Successfully: %s ('temperature': 100, 'humidity': 100)
Published data Successfully: %s ('temperature': 60, 'humidity': 14)
Published data Successfully: %s ('temperature': -19, 'humidity': 60)
Published data Successfully: %s ('temperature': 118, 'humidity': 53)
Published data Successfully: %s ('temperature': 6, 'humidity': 79)
```

status ("temperature":100,"humidity":100) json status 0 Simulations running

IBM WATSON OUTPUT:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area shows a device named '43127device' with a status of 'Disconnected' and a device type of '01101devicestype'. The 'Recent Events' tab is selected, displaying a table of events. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The events are all 'status' events with values representing temperature and humidity data. The bottom of the interface shows 'Items per page: 50' and '1-1 of 1 item'. A status bar at the bottom right indicates '0 Simulations running'.

Event	Value	Format	Last Received
status	{"temperature":92,"humidity":22}	json	a few seconds ago
status	{"temperature":1,"humidity":73}	json	a few seconds ago
status	{"temperature":6,"humidity":79}	json	a few seconds ago
status	{"temperature":128,"humidity":53}	json	a few seconds ago
status	{"temperature":10,"humidity":40}	json	a few seconds ago