

PYTHON CODE TO PUBLISH DATA TO IBM CLOUD

Python Code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
organization = "0vvv7i"
deviceType = "12345"
deviceId = "12"
authMethod = "token"
authToken = "12345678"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led is off")
    else:
        print ("Please send proper command")

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device %s" % str(e))
    sys.exit()

deviceCli.connect()

while True:
    temp=random.randint(90,110)
    Humid=random.randint(60,100)
    data = {'temp' : temp,'Humid' :Humid}
    def myOnPublishCallback():
        print("Published Temperature = %s C" % temp, "Humidity =%s %%" % Humid, "to IBM Watson")
```

```

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)

if not success:
    print("Not connected to IoTF")
    time.sleep(10)

deviceClid.commandCallback=myCommandCallback

deviceCli.disconnect()

```

The screenshot shows a Windows desktop environment with two windows open:

- Code Editor Window:** The title bar says "gasleakage.py - C:/Users/User/Desktop/gasleakage.py (3.7.0)". The content of the file is the Python code provided in the question.
- Python Shell Window:** The title bar says "Python 3.7.0 Shell". The output shows the script running and publishing data to IBM Watson.

```

gasleakage.py - C:/Users/User/Desktop/gasleakage.py (3.7.0)
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
organization = "0vvv7i"
deviceType = "12345"
deviceId = "12"
authMethod = "token"
authToken = "12345678"
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led is off")
    else:
        print ("Please send proper command")
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device %s" % str(e))
    sys.exit()
deviceCli.connect()
while True:
    temp=random.randint(90,110)
    Humid=random.randint(60,100)
    data = {'temp' : temp, 'Humid' : Humid}
    def myOnPublishCallback():
        print("Published Temperature = %s C" % temp, "Humidity =%s %%" % Humid, "to IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(10)
deviceClid.commandCallback=myCommandCallback
deviceCli.disconnect()

Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/User/Desktop/gasleakage.py =====
2022-11-18 12:00:05,573 ibmiotf.device.Client     INFO  Connected successfully: d:0vvv7i:12345:12
=====
RESTART: C:/Users/User/Desktop/gasleakage.py =====
2022-11-18 12:01:06,306 ibmiotf.device.Client     INFO  Connected successfully: d:0vvv7i:12345:12
Published Temperature = 96 C Humidity =84 % to IBM Watson
Published Temperature = 100 C Humidity =90 % to IBM Watson

```

Watson Cloud IBM:

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes links for Apps, LinkedIn, IBM, Tinkercad, GitHub, and cloud ibm. The user is logged in as mahashree4189@gmail.com with ID 0vv7i.

The main content area is titled "Device Drilldown - 12". On the left, a sidebar lists various device-related sections: Device Credentials, Connection Information, Recent Events (selected), State, Device Information, Metadata, Diagnostics, Connection Logs, and Device Actions. The "Recent Events" section displays a table of four entries from an IoT Sensor, each showing a timestamp of "a few seconds ago".

Event	Value	Format	Last Received
IoTSensor	{"temp":95,"Humid":81}	json	a few seconds ago
IoTSensor	{"temp":99,"Humid":72}	json	a few seconds ago
IoTSensor	{"temp":100,"Humid":90}	json	a few seconds ago
IoTSensor	{"temp":96,"Humid":84}	json	a few seconds ago

The "State" section shows a table with one entry: "3 Simulations running". At the bottom, there are two tabs: "python-3.7.0-amd64.exe" and "ibm_code.py.zip".