

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 Exceptions 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 IoTTF")

            time.sleep(10)
deviceCli.commandCallback=myCommandCallback
deviceCli.disconnect()

```

The screenshot shows a Python IDE window titled 'gasleakage.py - C:/Users/User/Desktop/gasleakage.py (3.7.0)' and a terminal window titled 'Python 3.7.0 Shell'.

The IDE window contains the following 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}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exceptions 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 IoTTF")
        time.sleep(10)
deviceCli.commandCallback=myCommandCallback
deviceCli.disconnect()

```

The terminal window shows the output of the script:

```

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 browser address bar displays the URL: `0vvv7linternetofthings.ibmcloud.com/dashboard/devices/drilldown/12345:12?returnTo=/devices/browse`. The top navigation bar includes the IBM Watson IoT Platform logo and a user profile for `mahashree4189@gmail.com` with ID `0vvv7i`.

The main content area is titled "Device Drilldown - 12" and features a left-hand sidebar with navigation options: Device Credentials, Connection Information, **Recent Events**, State, Device Information, Metadata, Diagnostics, Connection Logs, and Device Actions. The "Recent Events" section is active, displaying a table of recent data points.

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

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

The "State" section is also visible, showing a table of data points reported by the device. The current state is "3 Simulations running".

The bottom of the image shows a Windows taskbar with the following applications: `python-3.7.0-amd64.exe` and `ibm_code.py.zip`.