

Publish Data to the IBM Cloud

Team ID	PNT2022TMID46042
Project Name	Smart Waste Management System For Metropolitan Cities

PYTHON SCRIPT

```
PythonScript.py - C:/Python/Python37/PythonScript.py (3.7.4)
File Edit Format Run Options Window Help

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "dluuhi"
deviceType = "SWMS"
deviceId = "6032"
authMethod = "token"
authToken = "311519106032"

# Initialize GPIO

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

    #print(cmd)

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()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11

    temp=random.randint(0,100)
    Humid=random.randint(0,100)

    data = { 'temp' : temp, 'Humid': Humid }
    #print data
    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 IoT")
        time.sleep(1)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

OUTPUT:

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:09359112e, 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\PythonScript.py =====
2022-11-06 18:04:52,909 ibmiotf.device.Client INFO Connected successfully: d:dluuh:SWMS:6032
Published Temperature = 73 C Humidity = 97 % to IBM Watson
Published Temperature = 29 C Humidity = 49 % to IBM Watson
Published Temperature = 22 C Humidity = 38 % to IBM Watson
Published Temperature = 38 C Humidity = 23 % to IBM Watson
Published Temperature = 62 C Humidity = 82 % to IBM Watson
Published Temperature = 96 C Humidity = 54 % to IBM Watson
Published Temperature = 93 C Humidity = 73 % to IBM Watson
Published Temperature = 25 C Humidity = 57 % to IBM Watson
Published Temperature = 67 C Humidity = 26 % to IBM Watson
Published Temperature = 98 C Humidity = 100 % to IBM Watson
Published Temperature = 92 C Humidity = 54 % to IBM Watson
Published Temperature = 6 C Humidity = 59 % to IBM Watson
Published Temperature = 97 C Humidity = 57 % to IBM Watson
Published Temperature = 64 C Humidity = 70 % to IBM Watson
Published Temperature = 38 C Humidity = 14 % to IBM Watson
Published Temperature = 6 C Humidity = 49 % to IBM Watson
Published Temperature = 59 C Humidity = 73 % to IBM Watson
Published Temperature = 57 C Humidity = 20 % to IBM Watson
Published Temperature = 3 C Humidity = 42 % to IBM Watson
Published Temperature = 19 C Humidity = 42 % to IBM Watson
Published Temperature = 68 C Humidity = 19 % to IBM Watson
Published Temperature = 10 C Humidity = 14 % to IBM Watson
Published Temperature = 32 C Humidity = 67 % to IBM Watson
```

DATA IN IBM CLOUD PLATFORM:

The screenshot shows the IBM Cloud IoT Platform console. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays details for a device with ID '6032', which is 'Connected' and belongs to the 'SWMS' organization. The 'Recent Events' tab is selected, showing a table of live data events. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. The events are all from an 'IoTSensor' and are in 'json' format, with values like '["temp":100,"Humid":49]' and 'a few seconds ago' for the last received time.

Event	Value	Format	Last Received
IoTSensor	["temp":100,"Humid":49]	json	a few seconds ago
IoTSensor	["temp":4,"Humid":28]	json	a few seconds ago
IoTSensor	["temp":80,"Humid":86]	json	a few seconds ago
IoTSensor	["temp":28,"Humid":79]	json	a few seconds ago
IoTSensor	["temp":36,"Humid":98]	json	a few seconds ago