

DEVELOP THE PYTHON SCRIPT

Date	12 November 2022
Team ID	PNT2022TMID01100
Project Name	Project – Signs with smart connectivity for better safety.

Develop a python script:

The image shows a Python script in a code editor and its output in a terminal window. The script is named `pubsub.py` and is located in `D:\IBM\pubsub.py (3.7.0)`. The script imports `time`, `sys`, `ibmiotf.application`, `ibmiotf.device`, and `random`. It provides IBM Watson Device Credentials for organization `"p916a2"`, device type `"Ibmcloud_1"`, device ID `"Ibmcloud_1id"`, auth method `"token"`, and auth token `"!V1m8f-Ms*in_EqF"`. It initializes GPIO and defines a `myCommandCallback` function that prints the command received and its status. The script then connects to the IBM Watson IoT Platform and publishes sensor data (temperature and humidity) to the cloud as events of type `"hello"` with value `"world"`. The terminal window shows the output of the script, displaying the command received and its status, and the sensor data published to the cloud.

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

#Provide your IBM Watson Device Credentials
organization = "p916a2"
deviceType = "Ibmcloud_1"
deviceId = "Ibmcloud_1id"
authMethod = "token"
authToken = "!V1m8f-Ms*in_EqF"

# 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}
    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
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
    temp=random.randint(0,100)
    humd=random.randint(0,100)
    #Publish Temperature and Humidity data to the cloud
    deviceCli.publishEvent("hello", "world", {"temp": temp, "humid": humd})
    time.sleep(10)
```

Publish data to the IBM cloud:

The image shows the IBM Watson IoT Platform dashboard. The dashboard displays the following information:

- Device ID:** Ibmcloud_1id
- Status:** Connected
- Device Type:** Ibmcloud_1
- Class ID:** Device
- Date Added:** 13 Oct 2022 14:20
- Descriptive Location:** (empty)

The dashboard also shows a table of recent events:

Event	Value	Format	Last Received
IoTSensor	{ "temp":56,"Humid":7 }	json	a few seconds ago
IoTSensor	{ "temp":8,"Humid":1 }	json	a few seconds ago
IoTSensor	{ "temp":88,"Humid":75 }	json	a few seconds ago
IoTSensor	{ "temp":89,"Humid":24 }	json	a few seconds ago
IoTSensor	{ "temp":13,"Humid":54 }	json	a few seconds ago