

# **Develop A Python Script To Publish And Subscribe To IBM IoT Platform**

<b>Project Title</b>	<b>SmartFarmer – IoT Enabled Smart Farming Application</b>
<b>Team ID</b>	<b>PNT2022TMID22142</b>
<b>Content</b>	<b>Develop A Python Script To Publish And Subscribe To IBM IoT Platform</b>

# Python Script:

RED NODE.py - F:/RED NODE.py (3.11.0)

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 = "382h93" #replace the ORG ID
deviceType = "OUT"#replace the Device type wi
deviceId = "1234"#replace Device ID
authMethod = "token"
authToken = "q?Sr)VfQ&QUm_q79wi" #Replace the authToken
# Initialize GPIO

#Receives Command from Node-red
def myCommandCallback(cmd):
    print ("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    elif status == "motoroff" :
        print ("motor is off")
    elif status == "motor30" :
        print ("motor is on for 30 minutes")

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)
    soilmoisture=random.randint(0,100)

    data = { 'temp' : temp, 'Humid': Humid, 'soilmoisture': soilmoisture }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid, "soilmoisture = %s %"
        %soilmoisture, "to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
        time.sleep(5)

deviceCli.commandCallback = myCommandCallback

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



Browse

Action

Device Types

Interfaces

Add Device +

>		1234	Disconnected	Noder	Device	24 Oct 2022 09:50
▼		Testdevice1	Connected	Testing	Device	11 Nov 2022 15:08 → ...

Identity

Device Information

Recent Events

State

Logs

X

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

Event	Value	Format	Last Received
IoTSensor	{"temp":59,"Humid":96,"soilmoisture":100}	json	a few seconds ago
IoTSensor	{"temp":26,"Humid":59,"soilmoisture":99}	json	a few seconds ago
IoTSensor	{"temp":74,"Humid":13,"soilmoisture":96}	json	a few seconds ago
IoTSensor	{"temp":79,"Humid":24,"soilmoisture":28}	json	a few seconds ago

IBM

IBM-EPBL/IBM-Project

Node-RED.pdf

IBM Watson IoT Platform

Node-RED : node-red-...

Node-RED Dashboard

IBM Cloud and Node-RED

Download file | iLovePC

IBM App Development

← → ↺

https://node-red-zelxz-2022-11-08.au-syd.mybluemix.net/red/#flow/60daaa1e194009d0

90%

☆

🔒

📄

📶

🔌

☰

Node-RED

Deploy

👤

☰

filter nodes

text custom builder

text to speech

text to speech custom builder

tone analyzer v3

dashboard

button

dropdown

switch

slider

numeric

text input

date picker

colour picker

form

text

gauge

chart

audio out

notification

ui control

template

Flow 1

Flow 2

msg.payload

temp

humidity

soil moisture

temperature

humidity

soil moisture

motor on

motor off

text input

IBM IoT

IBM IoT OUT

debug

selected nodes

all

msg : string[21]

"JSON Message expected"

13/11/2022, 7:10:39 pm node: bf962d7744aaa484

msg.payload : Object

{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 ... }

13/11/2022, 7:11:13 pm node: bf962d7744aaa484

msg.payload : Object

{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 ... }

13/11/2022, 7:12:13 pm node: bf962d7744aaa484

msg.payload : Object

{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 ... }

13/11/2022, 7:13:13 pm node: bf962d7744aaa484

msg.payload : Object

{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 ... }

13/11/2022, 7:14:13 pm node: bf962d7744aaa484

msg.payload : Object

{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 ... }

13/11/2022, 7:15:13 pm node: bf962d7744aaa484

msg.payload : Object

{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 ... }

13/11/2022, 7:16:13 pm node: bf962d7744aaa484

msg.payload : Object

{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 ... }

13/11/2022, 7:18:55 pm node: IBM IoT OUT

msg : string[21]

https://node-red-zelxz-2022-11-08.au-syd.mybluemix.net/red/#

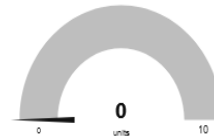
## Smart Agriculture System

Welcome!!

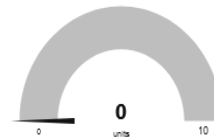
soil moisture



temprature



humidity



7c98b456c8900513

MOTOR ON

MOTOR OFF

## OUTPUT:

```
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\charu\Downloads\ibmiotpublishsubscribe.py =====
2022-11-11 15:56:49,907  ibmiotf.device.Client      INFO      Connected successfully: d:x0fxss:Testing:Testdevice1
Published Temperature = 8 C Humidity = 44 % soilmoisture = 3 % to IBM Watson
Published Temperature = 13 C Humidity = 95 % soilmoisture = 43 % to IBM Watson
Published Temperature = 78 C Humidity = 83 % soilmoisture = 83 % to IBM Watson
Published Temperature = 100 C Humidity = 52 % soilmoisture = 60 % to IBM Watson
Published Temperature = 45 C Humidity = 93 % soilmoisture = 16 % to IBM Watson
Published Temperature = 53 C Humidity = 12 % soilmoisture = 59 % to IBM Watson
Published Temperature = 15 C Humidity = 49 % soilmoisture = 32 % to IBM Watson
Published Temperature = 37 C Humidity = 73 % soilmoisture = 25 % to IBM Watson
```