

DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSCRIBE TO IBM IOT PLATFORM

TITLE	Smart Farmer-IOT Enabled Smart Farming Application
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID23830
Project Name	Smart Farmer - IoT Enabled Smart Farming Application
Leader Name	GOWSALYA L
Team Members Name	DEEPIKA B K MEGAVARSHINI G MONISHA N
MENTOR NAME	THIRUPPATHI M

PYTHON CODE:

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

```
#Provide your IBM Watson Device Credentials
organization = "kua3hx"
deviceType = "NodeMcu123"
deviceId = "12345"
authMethod = "token"
authToken = "1234567890"
```

```
#Initialize GPIO
```

```
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="Motor ON":
        print ("Motor is ON")
    else :
        print ("Motor 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)
    Moist=random.randint(0,100)

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

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTf")
        time.sleep(1)

    deviceCli.commandCallback = myCommandCallback
#Disconnect the device and the application from the cloud
deviceCli.disconnect()

```

```
IoT Smart Farming.py - D:/Program Files (x86)/Python/Python37/IoT Smart Farming.py (3.7.0)
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 = "kuazhix"
deviceType = "NodeMcu123"
deviceId = "12345"
authMethod = "token"
authToken = "1234567890"

#Initialize GPIO

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

    #print(cmd)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #.....

Ln: 47 Col: 72
```

```
IoT Smart Farming.py - D:/Program Files (x86)/Python/Python37/IoT Smart Farming.py (3.7.0)
File Edit Format Run Options Window Help
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)
    Moist=random.randint(0,100)

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

    success = deviceCli.publishEvent("IoT Sensor", "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 the application from the cloud
deviceCli.disconnect()

Ln: 47 Col: 72
```

PUBLISHING AND SUBSCRIBING TO IBM IOT PLATFORM:

The screenshot displays the IBM Watson IoT Platform dashboard. The main view shows a list of devices, with one device selected and its details visible. A modal window is open for configuring a new event type for the selected device.

Device Type: NodeMcu123

Events: 1

Event type name: event_1

Schedule: 20 Every Minute

Payload: Specify the event payload in the editor window or by uploading a CSV file.

Payload Editor:

```
0 {  
1   "Temperature": random(90, 110),  
2   "Humidity": random(60, 100),  
3   "Moisture": random(0, 100),  
4 }  
5
```

Recent Events Table:

Event	Value
event_1	{"Temperature":96,"Humidity":67,"Moisture":13}
event_1	{"Temperature":101,"Humidity":69,"Moisture":96}
event_1	{"Temperature":91,"Humidity":91,"Moisture":82}
event_1	{"Temperature":102,"Humidity":63,"Moisture":27}
event_1	{"Temperature":99,"Humidity":91,"Moisture":95}

The screenshot shows a Python script running in a terminal window, outputting data to IBM Watson IoT Platform. The output indicates a successful connection and the publishing of multiple events.

```
2022-11-15 01:30:59,188 ibmiotf.device.Client INFO Connected successfully: d:kua3hx:NodeMcu123:12345  
Published Temperature = 83 C Humidity = 38 % Moisture = 13 % to IBM Watson  
Published Temperature = 51 C Humidity = 16 % Moisture = 96 % to IBM Watson  
Published Temperature = 75 C Humidity = 27 % Moisture = 82 % to IBM Watson  
Published Temperature = 67 C Humidity = 48 % Moisture = 27 % to IBM Watson  
Published Temperature = 23 C Humidity = 40 % Moisture = 40 % to IBM Watson  
Published Temperature = 69 C Humidity = 49 % Moisture = 27 % to IBM Watson  
Published Temperature = 55 C Humidity = 11 % Moisture = 95 % to IBM Watson  
Published Temperature = 42 C Humidity = 100 % Moisture = 66 % to IBM Watson  
Published Temperature = 93 C Humidity = 9 % Moisture = 92 % to IBM Watson  
Published Temperature = 88 C Humidity = 20 % Moisture = 78 % to IBM Watson  
Published Temperature = 54 C Humidity = 54 % Moisture = 28 % to IBM Watson  
Published Temperature = 23 C Humidity = 22 % Moisture = 64 % to IBM Watson  
Published Temperature = 11 C Humidity = 41 % Moisture = 53 % to IBM Watson  
Published Temperature = 74 C Humidity = 42 % Moisture = 52 % to IBM Watson  
Published Temperature = 96 C Humidity = 28 % Moisture = 44 % to IBM Watson
```