

# **Smart Farmer - IOT Enabled Smart Farming Application**

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

<b>Team ID</b>	PNT2022TMID51719
<b>Team Leader</b>	SUREKHA S.K
<b>Team Members</b>	DENSIYA .I FENILDA RIJU R.F NISHA M.N

## **CODE:**

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

```
#Provide your IBM Watson Device Credentials
organization = "1jk4ps"
deviceType = "PNT2022TMID51719"
deviceId = "Smart_Farmer"
authMethod = "token"
```

```
authToken = "l1*53hClhmEbf!&Es&"
```

```
# 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("SDFRN", "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:

The image shows a Python script running in an IDE (IDLE Shell 3.9.8) and its output in a terminal. The script is a Python 3.9.8 application that connects to the IBM Watson IoT Platform and sends sensor data to the cloud.

**Python Script (ibm.py):**

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

# Provide your IBM Watson Device Credentials
organization = "1jk4ps"
deviceType = "PNT2022TMD51719"
deviceId = "Smart_Farmer"
authMethod = "token"
authToken = "11*53bC1h8Bf14E6s4"

# 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)
```

**Terminal Output:**

```
Python 3.9.8 (tags/v3.9.8:bb3fddc, Nov 5 2021, 20:48:33) [MSC v.1929 64 bit (AMD64)] on win
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Rija/OneDrive/Desktop/ibm.py python code.py =====
2022-11-11 15:19:18,051 ibmiotf.device.Client INFO Connected successfully: d:1jk4p
Published Temperature = 40 C Humidity = 86 % to IBM Watson
Published Temperature = 1 C Humidity = 61 % to IBM Watson
Published Temperature = 95 C Humidity = 51 % to IBM Watson
Published Temperature = 66 C Humidity = 54 % to IBM Watson
Published Temperature = 92 C Humidity = 10 % to IBM Watson
```

**IBM Watson IoT Platform Dashboard:**

The dashboard shows the device **Smart\_Farmer** (PNT2022TMD51719) is connected. The **Recent Events** tab displays a stream of data points:

Event	Value	Format	Last Received
SDFRN	{"temp":56,"Humid":69}	json	a few seconds ago
SDFRN	{"temp":46,"Humid":47}	json	a few seconds ago
SDFRN	{"temp":14,"Humid":19}	json	a few seconds ago
SDFRN	{"temp":99,"Humid":44}	json	a few seconds ago
SDFRN	{"temp":13,"Humid":18}	json	a few seconds ago

Items per page 50 | 1-3 of 3 items

0 Simulations running