

Smart Farmer – IoT Enabled Smart Farming

Application

FINAL CODE

Team ID	PNT2022TMID12127
Team Leader	AKSHAYAA RAAJA SHRI I T
Team Members	DHURGADEVI S KAMALEE S RAHUL B

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 displays a development environment with two windows. The left window shows a Python script for connecting to the IBM Watson IoT Platform. The right window shows the output of the script, which includes a REST API call and a successful connection message. Below the code windows, a browser window shows the IBM Watson IoT Platform interface, displaying a table of recent events for a device named 'Smart_Farmers'.

```
#!/usr/bin/env python
# coding: utf-8

import sys
import argparse
import json
import requests
import time

# Provide your IBM Watson Device Credentials
organization = "1234567890"
deviceid = "Smart_Farmers"
apikey = "1234567890"

# Initialize MQTT
mqtt = mqttclient.MQTTClient()

def main():
    # Connect to the MQTT broker
    mqtt.connect("wapi.1234567890.iot.cloud.ibm.com", 8883)

    # Publish a message
    mqtt.publish("Smart_Farmers", "Hello World")

    # Wait for a response
    time.sleep(10)

    # Disconnect
    mqtt.disconnect()

if __name__ == "__main__":
    main()
```

```
python 1.8.8 stage/v1.0.0/iotpython.py --org 1234567890 --deviceid Smart_Farmers --apikey 1234567890
REST API: 200 OK
Device: Smart_Farmers
Published Temperature = 40.0 Humidity = 66.9 to IBM Watson
Published Temperature = 41.0 Humidity = 61.9 to IBM Watson
Published Temperature = 42.0 Humidity = 51.9 to IBM Watson
Published Temperature = 43.0 Humidity = 54.9 to IBM Watson
Published Temperature = 44.0 Humidity = 50.9 to IBM Watson
```

IBM Watson IoT Platform

Smart_Farmers Connected FWI3022TMD001731 Device 11 Nov 2022 14:46

Event	Value	Format	Last Received
SDPFB	["temp":55,"humid":65]	json	a few seconds ago
SDPFB	["temp":45,"humid":42]	json	a few seconds ago
SDPFB	["temp":35,"humid":33]	json	a few seconds ago
SDPFB	["temp":55,"humid":44]	json	a few seconds ago
SDPFB	["temp":33,"humid":38]	json	a few seconds ago

Items per page: 50 | 1-5 of 5 items

0 Simulations running