

Sprint- 2

Team ID	PNT2022TMID11481
Project Title	Smart Farmer -IoT Enabled Smart Farming Application
Date	14.11.2022

IBM Watson and Python Integration:

By using Watson IoT Platform, you can collect connected device data and perform analytics on real-time data. The IBM Watson IoT Platform is a fully managed, Cloud-hosted service that provides device management capabilities as well as data collection and management in a time series format.



Your device or gateway

Start with your device and connect it with an IBM Cloud recipe.



MQTT and HTTP

Connect to the IBM Cloud using open, lightweight MQTT or HTTP.



IBM Watson® IoT Platform

Manage connected devices so your apps can access live and historical data.



REST and real-time APIs

Use highly-secure APIs to connect your apps with data from your devices.



Your application and analytics

Create analytic apps in the IBM Cloud, another cloud or your own servers.

Using the Device Created in IBM Watson:

IBM Watson IoT Platform

910619106003@smartinternz.com
ID: 4712i8

Browse Action Device Types Interfaces

Add Device +

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator ☒

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added
> <input type="checkbox"/>	2005	● Connected	akk	Device	1 Nov 2022 10:11

Items per page 50 | 1-1 of 1 item

0 Simulations running

IBM Watson IoT Platform

910619106003@smartinternz.com
ID: 4712i8

Browse Action Device Types Interfaces

Add Device +

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added	
▼ <input checked="" type="checkbox"/>	2005	● Connected	akk	Device	1 Nov 2022 10:11	→ ...

Identity Device Information Recent Events State Logs

Device ID 2005

Device Type akk

Date Added 1 Nov 2022 10:11

Added By 910619106003@smartinternz.com

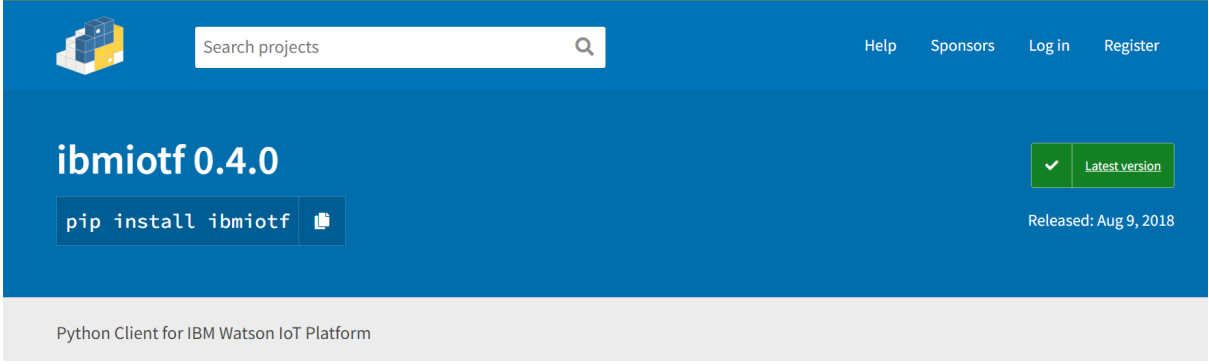
Connection Status **Connected**
Connection Time: 14 Nov 2022 00:26
Client Address: 49.204.141.9 SecureToken

Items per page 50 | 1-1 of 1 item

0 Simulations running

Connected sign shows that it is connected and live

Python code execution:



The screenshot shows the PyPI page for the `ibmiotf` package, version 0.4.0. The page has a blue header with the package name and version, a search bar, and links for Help, Sponsors, Log in, and Register. Below the header, there is a green button with a checkmark and the text "Latest version". A dark blue button with the text "pip install ibmiotf" and a document icon is also visible. The release date "Released: Aug 9, 2018" is shown in the bottom right. The footer of the page states "Python Client for IBM Watson IoT Platform".

Install this package : Python Client for IBM Watson IoT Platform

Python code:

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

#Provide your IBM Watson Device Credentials
organization = "47I2i8"
deviceType = "akk"
deviceId = "2005"
authMethod = "token"
authToken = "akk12345"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="switchon":
        print ("Switch is on")
    else :
        print ("Switch 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)
    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(1)

    deviceCli.commandCallback = myCommandCallback

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

```

```
IBMSmartFarmer.py - C:\Users\arao1\Desktop\IBMSmartFarmer.py (3.7.4)
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 = "4712i8"
deviceType = "akk"
deviceId = "2005"
authMethod = "token"
authToken = "akk12345"

# Initialize GPIO

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

Ln: 12 Col: 21

```
IBMSmartFarmer.py - C:\Users\arao1\Desktop\IBMSmartFarmer.py (3.7.4)
File Edit Format Run Options Window Help

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" 1
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)

    success = deviceCli.publishEvent("IoTSensor", "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()
```

Ln: 12 Col: 21

```
IBMsmartFarmer.py - CAU
File Edit Format Run
print
sys.ex
# Connect and
deviceCli.con
while True:
    #Get s
    temp=
    Humid=
    SoilMo
    data =
    #print
    def my
    pr
    succes
    if not
    pr
    time.s
    device
# Disconnect t
deviceCli.disc

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\araol\Desktop\IBMsmartFarmer.py =====
2022-11-14 00:45:30,852 ibmiotf.device.Client INFO Connected successfu
lly: d:4712i8:akk:2005
Published Temperature = 97 C Humidity = 23 % SoilMoisture = 80 % to IBM Watson
Published Temperature = 25 C Humidity = 2 % SoilMoisture = 24 % to IBM Watson
Published Temperature = 45 C Humidity = 42 % SoilMoisture = 34 % to IBM Watson
Published Temperature = 35 C Humidity = 20 % SoilMoisture = 31 % to IBM Watson
Published Temperature = 92 C Humidity = 98 % SoilMoisture = 25 % to IBM Watson
Published Temperature = 29 C Humidity = 26 % SoilMoisture = 98 % to IBM Watson
eeting" 1
ure = %s :
hCallback
Ln: 12 Col: 2
```

Recent Events in IBM Watson IoT Platform:

IBM Watson IoT Platform					
Browse Action Device Types Interfaces Add Device +					
2005 Connected akk Device 1 Nov 2022 10:11 → ...					
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":19,"Humid":50,"SoilMoisture":45}	json	a few seconds ago		
IoTSensor	{"temp":80,"Humid":50,"SoilMoisture":94}	json	a few seconds ago		
IoTSensor	{"temp":40,"Humid":86,"SoilMoisture":91}	json	a few seconds ago		
IoTSensor	{"temp":50,"Humid":55,"SoilMoisture":100}	json	a few seconds ago		
IoTSensor	{"temp":47,"Humid":37,"SoilMoisture":29}	json	a few seconds ago		

Boards in IBM Platform:

