

# **SPRINT 1**

## **TEAM ID: PNT2022TMID49415**

### **Smart Waste Management System For Metropolitan Cities**

#### **PYTHON CODE:**

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "t5udfe"
deviceType = "raspberrypi"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
# Initialize GPIO
def myCommandCallback(cmd):
    print('Command received: %s' % cmd.data['command'])
    status=cmd.data['command']
    if status=="smart bin opened":
        print ("The Smart Bin is Open now")
    else :
        print ("The Smart Bin is Close now")
    #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**

**distance=random.randint(0,200)**

**weight=random.randint(0,10)**

**data = { 'distance' : distance, 'weight': weight }**

**#print data**

**def myOnPublishCallback():**

**print ("Published Data to IOT Watson: \n      Distance= %s cm\n" % distance, "      Weight = %s Kg\n" % weight)**

**success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback)**

**if not success:**

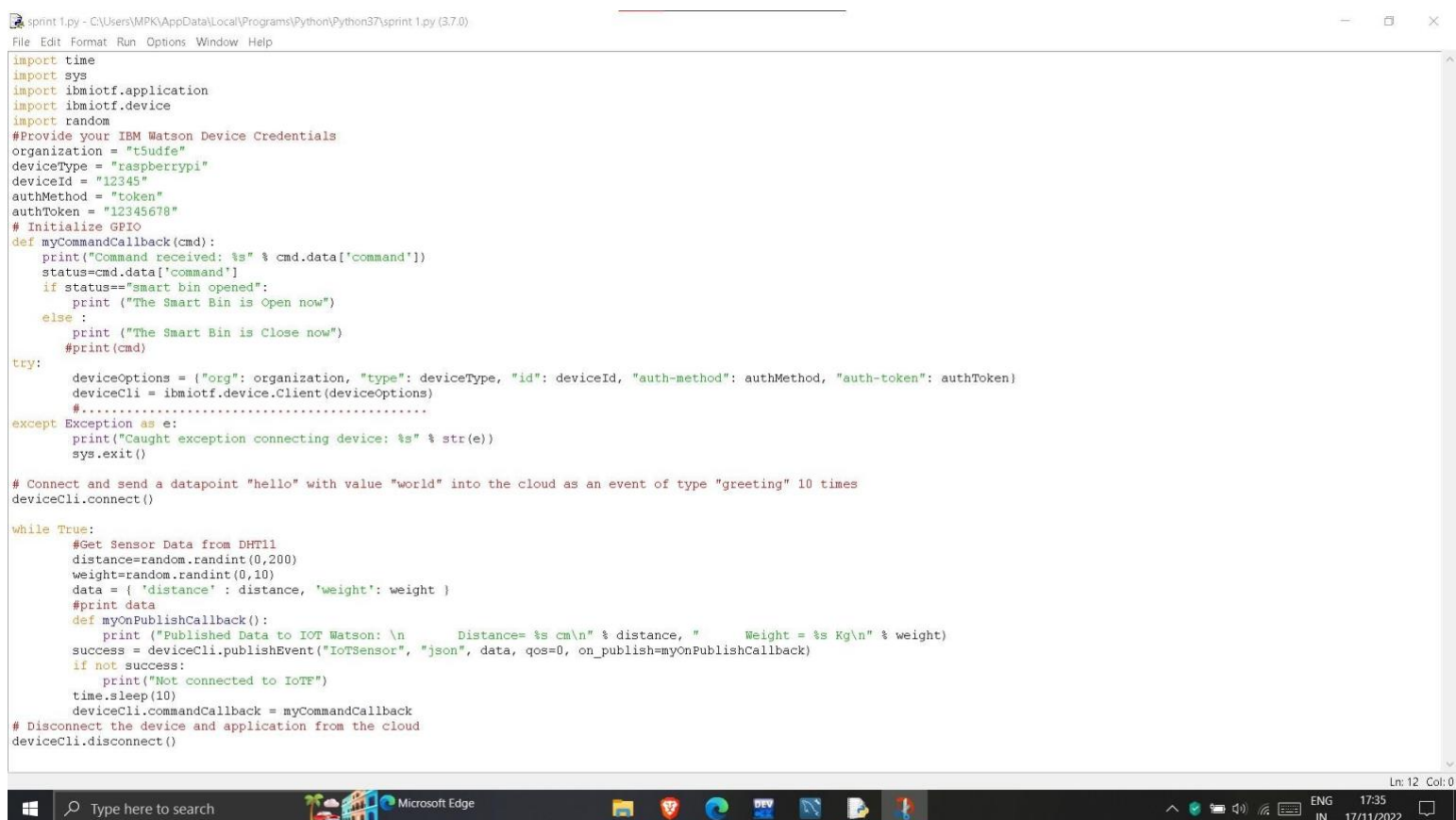
**print("Not connected to IoTF")**

**time.sleep(10)**

**deviceCli.commandCallback = myCommandCallback**

**# Disconnect the device and application from the cloud**

**deviceCli.disconnect()**



The screenshot shows a code editor window titled "sprint 1.py" with a menu bar (File, Edit, Format, Run, Options, Window, Help) and a toolbar. The code is a Python script for connecting to IBM Watson IoT and publishing sensor data. It includes imports for time, sys, ibmiotf, and random. It defines a myCommandCallback function and a myOnPublishCallback function. The main logic is in a while True loop that generates random distance and weight values, publishes them to the cloud, and handles connection errors. The script ends with a disconnect call. The status bar at the bottom shows "Ln: 12 Col: 0".

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

#Provide your IBM Watson Device Credentials
organization = "tSudfe"
deviceType = "raspberrypi"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="smart bin opened":
        print ("The Smart Bin is Open now")
    else :
        print ("The Smart Bin is Close now")
    #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
    distance=random.randint(0,200)
    weight=random.randint(0,10)
    data = { 'distance' : distance, 'weight': weight }
    #print data
    def myOnPublishCallback():
        print ("Published Data to IOT Watson: \n      Distance= %s cm\n" % distance, "      Weight = %s Kg\n" % weight)
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(10)
    deviceCli.commandCallback = myCommandCallback

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

The Python code for generating random values for the parameters.

```
Python 3.7.0 Shell*
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\MPK\AppData\Local\Programs\Python\Python37\sprint 1.py ==
2022-11-17 17:41:19,440 ibmiotf.device.Client INFO Connected successfully: d:t5udfe:raspberrypi:12345
Published Data to IOT Watson:
    Distance= 48 cm
    Weight = 8 Kg
Published Data to IOT Watson:
    Distance= 168 cm
    Weight = 2 Kg
Published Data to IOT Watson:
    Distance= 198 cm
    Weight = 10 Kg
Published Data to IOT Watson:
    Distance= 113 cm
    Weight = 5 Kg
Published Data to IOT Watson:
    Distance= 143 cm
    Weight = 3 Kg
Published Data to IOT Watson:
    Distance= 165 cm
    Weight = 9 Kg
Published Data to IOT Watson:
    Distance= 159 cm
    Weight = 10 Kg
Published Data to IOT Watson:
    Distance= 135 cm
    Weight = 3 Kg
Ln: 5 Col: 0
```

Here we are generating random values for both the parameters weight and distance with the help of the random function in python. The weight parameter denotes the weight of smartbin and the distance parameter denotes the amount of garbage present in the smartbin which has a maximum length of 200 cm.

IBM Watson IoT Platform

Browse Action Device Types Interfaces

## 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

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	raspberrypi	Device	12 Nov 2022 19:26	

Items per page 50 | 1-1 of 1 item

1 Simulation running

IBM Watson IoT Platform

Browse Action Device Types Interfaces

## Recent Events

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

Event	Value	Format	Last Received
IoTSensor	{"distance":153,"weight":1}	json	a few seconds ago
IoTSensor	{"distance":29,"weight":2}	json	a few seconds ago
IoTSensor	{"distance":173,"weight":1}	json	a few seconds ago
IoTSensor	{"distance":10,"weight":9}	json	a few seconds ago
IoTSensor	{"distance":131,"weight":7}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

0 Simulations running

A new device is created and the random values from the python code is connected to the iot sensors. These random values are considered to be sensor values