

SPRINT 1

TEAM ID	PNT2022TMID33826
PROJECT NAME	Smart Waste Management System For Metropolitan Cities

PYTHON CODE:

```
Finalproject.py - C:/Users/ADMIN/Downloads/Finalproject.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 = "3f3tah"
deviceType = "sensor"
deviceId = "123456"
authMethod = "token"
authToken = "1234567890"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="binfull":
        print ("-----EMPTY THE BIN IMMEDIATELY-----")

    #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()

Ln: 42 Col: 0
```

BEFORE CONNECTED TO IBM WATSON:

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The 'Browse Devices' section has tabs for 'All Devices' and 'Diagnose'. A message states: '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.' Below this is a search bar 'Search by Device ID' and a 'Device Simulator' toggle switch. The table below shows one device with the status 'Disconnected'.

Device ID	Status	Device Type	Class ID	Date Added
123456	Disconnected	sensor	Device	Nov 13, 2022 9:38 AM

Items per page 50 | 1-1 of 1 item

1 Simulation running

Activate Windows
Go to Settings to activate Windows.

AFTER CONNECTED TO IBM WATSON:

The screenshot shows the same IBM Watson IoT Platform interface, but the device status has changed to 'Connected'.

Device ID	Status	Device Type	Class ID	Date Added
123456	Connected	sensor	Device	Nov 13, 2022 9:38 AM

Items per page 50 | 1-1 of 1 item

1 Simulation running

Activate Windows
Go to Settings to activate Windows.

The screenshot shows a Python 3.7.0 Shell window with the following code and output:

```

import time
import sys
import ibmiotf
import random

#Provide your organization
deviceType = "device"
deviceId = "1"
authMethod = "token"
authToken = "1234567890"

# Initialize

def myCommand(status="cm"):
    print("Command status: " + status)
    if status == "cm":
        print("BIN IS IN NORMAL LEVEL")

    #print(cmd)

try:
    device = ibmiotf.device.Client(deviceId, deviceType, authMethod, authToken)
    #...

except Exception as e:
    print(e)
    sys.exit(1)

# Connect and deviceCli.com

```

Output:

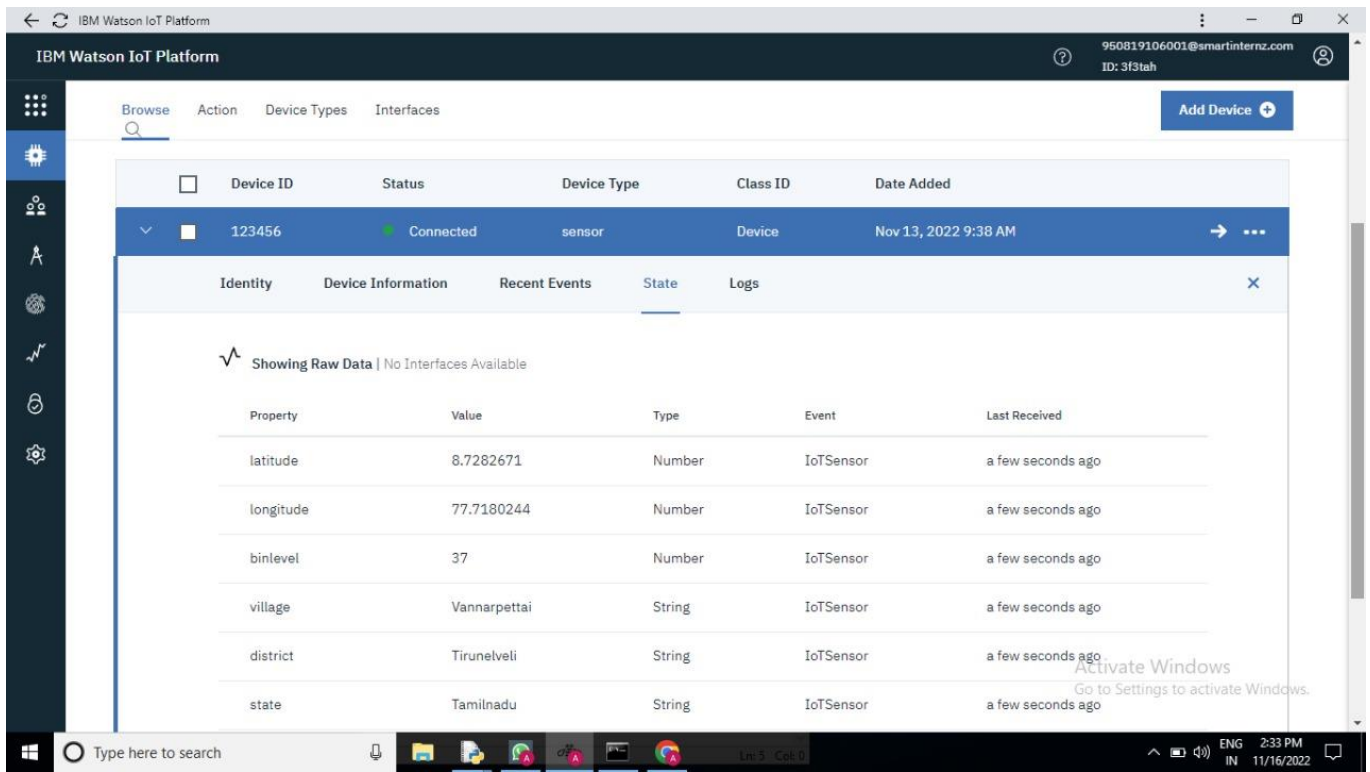
```

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/ADMIN/Downloads/Finalproject.py =====
BIN IS IN NORMAL LEVEL
2022-11-16 14:32:37,885 ibmiotf.device.Client INFO Connected successfully: d:3f3tah:sensor:123456
Published Latitude = 8.7289086 Longitude = 77.6745726 % Binlevel = 54 village = Nellai Town district= Tirunelveli state = Tamilnadu country = India to IBM Watson
BIN IS IN NORMAL LEVEL
Published Latitude = 8.7289086 Longitude = 77.6745726 % Binlevel = 56 village = Nellai Town district= Tirunelveli state = Tamilnadu country = India to IBM Watson
BIN IS IN NORMAL LEVEL

```

[illegible]

STATE:



The screenshot displays the IBM Watson IoT Platform web interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present, and an 'Add Device' button is in the top right. The main content area shows a table of devices. The selected device has ID '123456', status 'Connected', type 'sensor', and class ID 'Device'. Below the table, the 'State' tab is active, showing raw data. The data is presented in a table with columns: Property, Value, Type, Event, and Last Received. The data includes latitude, longitude, binlevel, village, district, and state, all reported 'a few seconds ago'. A watermark 'Activate Windows' is visible in the bottom right corner of the interface.

Device ID	Status	Device Type	Class ID	Date Added
123456	Connected	sensor	Device	Nov 13, 2022 9:38 AM

Property	Value	Type	Event	Last Received
latitude	8.7282671	Number	IoTSensor	a few seconds ago
longitude	77.7180244	Number	IoTSensor	a few seconds ago
binlevel	37	Number	IoTSensor	a few seconds ago
village	Vannarpettai	String	IoTSensor	a few seconds ago
district	Tirunelveli	String	IoTSensor	a few seconds ago
state	Tamilnadu	String	IoTSensor	a few seconds ago

PYTHON CODE:

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

```
#Provide your IBM Watson Device Credentials
organization = "3f3tah"
deviceType = "sensor"
deviceId = "123456"
authMethod = "token"
authToken = "1234567890"
```

```
# Initialize GPIO
```

```
def myCommandCallback(cmd):
    print("Command received: %s" %
cmd.data['command'])
    status=cmd.data['command']
    if status=="binfull":
        print ("-----EMPTY THE BIN
IMMEDIATELY----")
```

```
#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
```

```
    binlevel=random.randint(10,100)
    locationId=random.randint(1,5)
    district="Tirunelveli"
    state="Tamilnadu"
    country="India"
    if locationId == 1:
        latitude=8.7060581
        longitude=77.7633162
        village="VM Chathiram"
    elif locationId == 2:
        latitude=8.7066676
        longitude=77.732578
        village="Perumalpuram"
    elif locationId == 3:
        latitude=8.7199159
```

```

        longitude=77.725674
        village="Palayamkottai"
    elif locationId == 4:
        latitude=8.7282671
        longitude=77.7180244
        village="Vannarpettai"
    elif locationId == 5:
        latitude=8.7289086
        longitude=77.6745726
        village="Nellai Town"
    else:
        print("No location Found!!")

    data = { 'latitude' : latitude, 'longitude':
longitude,'binlevel':
binlevel,'village':village,'district':district,'state':stat
e,'country':country }
    #print data
    def myOnPublishCallback():
        print ("Published Latitude = %s " %
latitude, "Longitude = %s %% " % longitude,
"Binlevel = %s" % binlevel,"village = %s " %
village,"district= %s" % district,"state = %s" %
state,"country = %s " % country, "to IBM
Watson\n")
        if binlevel >= 90:
            data={'Latitude':latitude,
'Longitude':longitude, 'Binlevel':binlevel,
'Village':village, 'District':district,
'State':state,'Country':country}

print("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!BIN IS
FULL
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!\n")
        print("-----
EMPTY THE BIN IMMEDIATELY-----
-----\n")
        deviceCli.commandCallback =
myCommandCallback
        time.sleep(5)
    else:
        print("BIN IS IN NORMAL LEVEL\n")
        time.sleep(5)

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

```

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