

## **PROJECT DEVELOPMENT**

### **PHASEPROJECT DEVELOPMENT DELIVERY**

#### **SPRINT-3**

Date	12 Nov 2022
Team ID	PNT2022TMID38568
Project Name	Smart Waste Management System For Metropolitan Cities
Marks	

#### **CODE EXPLANATION**

The below described code is what we have developed for connecting with IBM IoT cloud. The code also connects with the Node- Red Service and displays the output frequently. Once the Code is simulated, the code runs with the output of temperature, humidity and pH value. This code links with the IBM IoT Platform and then to Node-Red, Finally the result is displayed in our Application.

#### **PYTHON CODE:**

```
import time
```

```
import sys
```

```
import ibmiotf.application
```

```
import ibmiotf.device
```

```
import random
```

```
#Thiyagarajan IBM
```

```
organization = "7gwfxT"
```

```
deviceType = "T1"
```

```
deviceId = "12345"
```

```
authMethod = "token"
```

```
authToken = "123456789"
```

```
#Gpio
```

```
def level():
```

```
    c=""
```

```
    if Bin>=90:
```

```
        c="The Bin is full "
```

```
    else:
```

```

        c="The Bin level is not full"

    return c

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()

#CONNECCT

deviceCli.connect()

while True:

    Bin=random.randint(0,100)

    a=level()

    def myOnPublishCallback():

        print("Published Bin level = %s %" %Bin,"Level = %s " %a,"to IBM Watson")

    data={'bin':Bin,'level':level()}

    success = deviceCli.publishEvent("IoTSensor","json",data,qos=0,
on_publish=myOnPublishCallback)

    if not success:

        print("Not connected to IoT")

    time.sleep(10)

```

**#Disconnect**

**deviceCli.disconnect()**