

DEVELOP THE PYTHON SCRIPT

Team ID	PNT2022TMID11896
Project Name	Smart Waste Management System For Metropolitan

```
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=="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:
TEAMID PNT2022TMID33826
PROJECT NAME Smart Waste Management
System For Metropolitan Cities
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("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()
```