

PUBLISH DATA TO THE IBM CLOUD

TEAMID	PNT2022TMID53817
PROJECT NAME	Smart Waste Management System For Metropolitan Cities

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 with the text 'Search by Device ID'. The main content area shows a table of devices. One device, with ID '123456', is listed as 'Connected' and is a 'sensor' type. Below the table, the 'Recent Events' tab is selected, showing a list of events. The events are listed in a table with columns 'Event', 'Value', and 'Format'. The events are IoTSensor events with values like ['temp':76,'Humid':44] and ['temp':84,'Humid':26]. A 'Python 3.7.0 Shell' window is open over the events table, displaying a stream of temperature and humidity data published to IBM Watson.

Event	Value	Format
IoTSensor	['temp':76,'Humid':44]	json
IoTSensor	['temp':84,'Humid':26]	json
IoTSensor	['temp':34,'Humid':85]	json
IoTSensor	['temp':91,'Humid':79]	json
IoTSensor	['temp':21,'Humid':38]	json

The screenshot shows a Python script named 'ibmiotpublishsubscribe.py' in a code editor. The script is designed to connect to the IBM Watson IoT Platform and publish data. It includes comments for providing IBM Watson Device Credentials, initializing the GPID, and a callback function for handling commands. The script uses the 'ibmiotf' library to connect and send a datapoint 'hello' with the value 'world' as an event of type 'greeting' 10 times.

```
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 GPID

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:
    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()
```

