

# PUBLISH DATA TO THE IBM CLOUD

TEAMID	PNT2022TMID32932
PROJECT NAME	Smart Waste Management System For Metropolitan Cities

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various IoT functions. The main content area shows the 'Recent Events' tab for a specific device. Below the tab, a message states: 'The recent events listed show the live stream of data that is coming and going from this device.' A table lists five recent events, each with a 'data' event type and a 'warning' value. The table columns are 'Event', 'Value', 'Format', and 'Last Received'. At the bottom of the dashboard, a status bar indicates '1 Simulation running'.

Event	Value	Format	Last Received
data	{"warning":52.4}	json	a few seconds ago
data	{"warning":22.9}	json	a few seconds ago
data	{"warning":180.5}	json	a few seconds ago
data	{"warning":198.4}	json	a few seconds ago
data	{"warning":122.8}	json	a few seconds ago

1 Simulation running

```
waste mana.py - C:\Users\Preethi Asok\AppData\Local\Programs\Python\Python311\waste mana.py (3.11.0)
File Edit Format Run Options Window Help

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

organizationID='1bdx6w'
deviceType='preethi'
deviceId='171122'
authMethod='token'
authToken='12345678'

def myCommandCallback(cmd):
    print("Command received: %s" %cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print('led is on')
    elif status=="lightoff":
        print('led is off')
    else:
        print('please send proper command')
try:
    deviceOption={"org":organizationID,"type":deviceType,"id":deviceId,"auth-method":authMethod,"auth-token":authToken}
    deviceCli = ibmiotf.device.Client(deviceOption)
except Exception as e:
    print("Caught exception connecting device: %s" %str(e))
sys.exit()
deviceCli.connect()
while True:
    temp=random.randint(90,100)
    Humid=random.randint(10,100)
    data ={'temp': temp,'Humid': Humid}
    def myOnPublishCallback():
        print("Published Distance=%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 IOTF");
    time.sleep(10)

deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()
```