

## Publish Data to IBM cloud

Team ID	PNT2022TMID31030
Project Name	Smart waste management system for metropolitan cities

### **Steps:**

**Step 1:** Open python idle.

**Step 2:** Type the program.

**Step 3:** Then click on the file and save the document.

**Step 4:** Then click on run and run the module.

**Step 5:** Output will be appeared in the idle window.

### **Python code:**

```
import time
```

```
import sys
```

```
import ibmiotf.device
```

```
import random
```

```
organization="gpx238"
```

```
deviceType="NodeMCU"
```

```
deviceId="123456"
```

```
authMethod="token"
```

```
authToken="12345678"
```

```
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()
```

```

deviceCli.connect()
while True:
    #in data
    latitude=11.7345;
    longitude=78.2020;

    #out data
    #latitude=12.7345;
    #longitude=79.2020;

    data={'latitude':latitude,'longitude':longitude}
    def myOnPublishCallback():
        print("published latitude=%d" %latitude,"longitude=%d" %longitude,"to ibm
watson")

success=deviceCli.publishEvent("IotSensor","json",data,qos=0,on_publish=myOnPublishCal
lback)
    if not success:
        print("Not connected to IoT")
        time.sleep(3)
deviceCli.disconnect()

```

The image shows a Windows desktop environment. At the top, a file explorer window is open, displaying the path 'C:\Users\91908\Documents\Public Data.py (3.11.0)'. Below this, a code editor window contains a Python script. The script imports modules like 'time', 'sys', 'ibmiotf.device', and 'random'. It defines variables for 'organization', 'deviceType', 'deviceId', 'authMethod', and 'authToken'. A 'try' block contains logic to create a 'deviceOptions' dictionary, instantiate a 'deviceCli' object, and attempt to connect to an IoT device. An 'except' block handles 'Exception' with a print statement and 'sys.exit()'. After the 'try' block, the script calls 'deviceCli.connect()', enters a 'while True' loop to send and receive data (latitude and longitude), and includes a 'success=deviceCli.publishEvent()' call. It also has a 'disconnect()' method call at the end. The bottom of the screen features a Windows taskbar with various application icons, including the Start button, File Explorer, Edge, and several instances of Google Chrome. The system tray on the right shows the temperature (78°F), network status, and the date/time (20:38, 15-11-2022).

[illegible]

### Output screen (Data published)