

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

```
#Provide your IBM Watson Device Credentials
```

```
organization = "6cpv77"
```

```
deviceType = "weather"
```

```
deviceId = "wea_today"
```

```
authMethod = "token"
```

```
authToken = "Fo9af9o00doS8VUsb2"
```

```
# Initialize GPIO
```

```
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="Alarm On":
        print("Alarm is off")
    else:
        print("Alarm is On")
    #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()
```

```
while True:
```

```
    #Get Sensor Data from DHT11
```

```
    temp=random.randint(0,200)
```

```
    humidity=random.randint(0,200)
```

```
    if temp>=100:
```

```
        print("Alert")
```

```
    else:
```

```
        print("safe")
```

```
data = {"d":{"temp" : temp, 'humidity': humidity}}
```

```
#print data
```

```
def myOnPublishCallback():
```

```
    print ("Published Temperature = %s C" % temp, "Humidity = %s %" % humidity, "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()
```