```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import ibmiotf.api
import requests
import json
#Provide your IBM Watson Device Credentials
organization = "lryrya"
deviceType = "Gas_Sensor"
deviceId = "G1"
authMethod = "token"
authToken = "123456789"
NO2=0
CO2=0
CO=0
alert=0
def Alert():
    url = "https://www.fast2sms.com/dev/bulkV2"
    my_data = {
            'sender_id': 'FSTSMS',
            'message': 'Alert some abnormal conditions were found in XYZ industry! Kindly
Evacuvate from the place',
            'language': 'english',
            'route': 'p',
            'numbers': '9095057479'
     }
    headers = {
          'authorization':
'o0azwVFNHOM5B3hrRxdenyU2cfZujqSpYEX7t8LAgJPb9kliWCugDvo1n0kcY8TGHOt3dIQws
KpLbAJU',
          'Content-Type': "application/x-www-form-urlencoded",
          'Cache-Control': "no-cache"
     }
    response = requests.request("POST",
                                                    url,
                                                    data = my_data,
                                                    headers = headers)
    returned_msg = json.loads(response.text)
    print(returned_msg['message'])

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()
while True:
   if(NO2<50) or (CO2<50) or (CO<50):
      #Get Sensor Data from DHT11
      NO2=NO2+1
      CO2=CO2+1
      CO=CO+1
      data = { 'NO2' : NO2 , 'CO2': CO2, 'CO' : CO}
      #print data
      def myOnPublishCallback():
         print ("Published Nitrogen di Oxide = %s %%" % NO2 ,"Carbon di oxide = %s %%" %
CO2 , "Carbon monoxide = %s %%" % CO, "to IBM Watson")

      if((NO2>25) or (CO2>25) or (CO>25)) and (alert==0):
         Alert();
         time.sleep(2)
         alert=alert+1
      success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
   else:
      NO2=2;
      CO2=3;
      CO=1;
   if not success:
      print("Not connected to IoTF")
   time.sleep(2)
deviceCli.disconnect()
```