import wiotp.sdk.device

import time

import random

import ibmiotf.application

import ibmiotf.device

import requests, json

myConfig = {

#Configuration

"identity": {

"orgId": "n6rl9n",

"typeId": "NodeMCU",

"deviceId":"621319106312"

},

#API Key

"auth": {

"token": "9876543210"

}

}

#Receiving callbacks from IBM IOT platform

def myCommandCallback(cmd):

print("Message received from IBM IoT Platform: %s" % cmd.data['command'])

m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)

client.connect()

#OpenWeatherMap Credentials

BASE\_URL = "https://api.openweathermap.org/data/2.5/weather?"

CITY = "Salem, IN"

URL = BASE\_URL + "q=" + CITY + "&units=metric"+"&appid=" + "f58e4720c739a54c439aba9b05176839"

while True:

response = requests.get(URL)

if response.status\_code == 200:

data = response.json()

main = data['main']

temperature = main['temp']

humidity = main['humidity']

pressure = main['pressure']

report = data['visibility']

#messge part

msg=random.randint(0,5)

if msg==1:

message="GO SLOW, SCHOOL ZONE AHEAD"

elif msg==2:

message="NEED HELP, POLICE STATION AHEAD"

elif msg==3:

message="EMERGENCY, HOSPITAL NEARBY"

elif msg==4:

message="DINE IN, RESTAURENT AVAILABLE"

elif msg==5:

message="PETROL BUNK NEARBY"

else:

message=""

#Speed Limit part

speed=random.randint(0,150)

if speed>=100:

speedMsg=" Limit Exceeded"

elif speed>=60 and speed<100:

speedMsg="Moderate"

else:

speedMsg="Slow"

#Diversion part

sign=random.randint(0,5)

if sign==1:

signMsg="Right Diversion"

elif sign==2:

signMsg="Speed Breaker"

elif sign==3:

signMsg="Left Diversion"

elif sign==4:

signmsg="U Turn"

else:

signMsg=""

#Visibility

if temperature < 24:

visibility="Fog Ahead, Drive Slow"

elif temperature < 20:

visibility="Bad Weather"

else:

visibility="Clear Weather"

else:

print("Error in the HTTP request")

myData={'Temperature':temperature, 'Message':message, 'Sign':signMsg, 'SpeedValue': speed, 'Speed': speedMsg, 'Visibility':visibility}

client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None) #PUBLISHING TO IOT WATSON

print("Published data Successfully: ", myData)

print("----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------")

client.commandCallback = myCommandCallback

time.sleep(5)

client.disconnect()