

```

import
wiotp.sdk
.device

#importing library files for connecting with CLOUD,sdk=software developement
kit
import requests
#for API request
import json
#converting it to json(key:values)
myConfig = {
    "identity": {
        "orgId": "uaortj",
        "typeId": "Monitor_devicetype",      #configuration wit CLOUD,finding
identity
        "deviceId":"Monitor_deviceid"
    },
    "auth": {
        "token": "sngs123monitor"    #authenticating with cloud device
    }
}

#TRAFFIC AND FATAL SITUATION ALERT MESSAGE DISPLAYING IN WEB UI WHWN THE
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
#initialising device client with above myconfig detail
client.connect()
def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" %cmd.data['command'])
    m=cmd.data['command']
    ALERT=""
#THIS IF CONDITION BLOCK IS FOR TRAFFIC AND FATAL SITUATION ALERT MESSAGE
DISPLAYING IN WEB UI WHEN THE MESSAGE WAS RECEIVED FROM THE ROAD SAFETY
OFFICE
    if (m=="TRAFFIC"):
        ALERT="TRAFFIC - TAKE DIVERSION"
        print("*****//TAKE DIVERSION//*****")
    elif(m=="ACCIDENT"):
        ALERT="ACCIDENT - TAKE DIVERSION"
        print("*****//TAKE DIVERSION//*****")
    else:
        ALERT="HAVE A NICE DAY!"
        print("HAVE A NICE DAY!")
        mydata1={"SITUATION":ALERT,}
        client.publishEvent("Monitor_deviceid","json",mydata1)
while True:
    print("=====")

```

```

weatherData =
requests.get('https://api.openweathermap.org/data/2.5/weather?q=Chennai,IN&a
ppid=b23b5fad240356d80f95242dcf1d6cad')
b = weatherData.json()
temp = b["main"]["temp"]
humi = b["main"]["humidity"]
main = b["weather"][0]["main"]      #0th index is taken from the
object
description = b["weather"][0]["description"]
visibility = b["visibility"]
Windspeed = b["wind"]["speed"]
TemperatureRecommendation = ""
SpeedRecommendation = ""
RecommendationForVisibilty = ""
#print("Temperature(celcius) :",b["main"]["temp"])
if(temp > 33):
    TemperatureRecommendation="Temperature is higher than ideal
value"
    print("Temperature is higher than ideal value")
elif(temp<19):
    TemperatureRecommendation="Temperature is lower than ideal
value"
    print("Temperature is lower than ideal value")
else:
    TemperatureRecommendation="Temperature is ideal"
    print("Temperature is ideal ")
#print("Humidity :",b["main"]["humidity"])
#print("WeatherCondition",(b["weather"][0]["main"]))
if(main == "Rain"):
    rain = b["rain"]["1h"]
    SpeedRecommendation = "30KM/HR ,ROAD WILL BE SLIPPERY"
#print("Rain:",b["rain"]["1h"])
#print("SPEED RECOMMENDATION : 30KM/HR ,ROAD WILL BE SLIPPERY")
elif(main == "Drizzle"):
    SpeedRecommendation = "30KM/HR"
#print("SPEED RECOMMENDATION : 30KM/HR")
elif(main == "Mist"):
    SpeedRecommendation = "30KM/HR and switch on the headlight"
#print("SPEED RECOMMENDATION : 30KM/HR and switch on the Headlight")
elif(main == "Thunderstorm"):
    SpeedRecommendation = "30KM/HR and stay away in the open place"
#print("SPEED RECOMMENDATION : 30KM/HR and stay away in the open place")
#print("Description of weather :",b["weather"][0]["description"])
#print("visibility",(b["visibility"]))
if(visibility<1000):

```

```

        RecommendationForVisibilty = "SPEED RECOMMENDATION : 30KM/HR and
SWITCH ON THE HEAD LIGHT"
    else:
        RecommendationForVisibilty = "Visibility range is ideal for
vechicles"
#print("SPEED RECOMMENDATION : 30KM/HR and SWITCH ON THE HEAD LIGHT")
    mydata={"temperature":temp,
"TemperatureRecommendation":TemperatureRecommendation,"humidity":humi,"Weath
erCondition":main,"SpeedRecommendation":SpeedRecommendation
,"DescriptionOfWeather":description,"visibility":visibility,"RecommendationF
orVis ibility":RecommendationForVisibilty,"WindSpeed":Windspeed}
    print(mydata)
    client.publishEvent("Monitor_deviceid","json",mydata)
    client.commandCallback = myCommandCallback

```