Packages to be installed :-

1.

pip install paho-mqtt

<https://pypi.org/project/paho-mqtt/>

**Flask Server:-**

from flask import Flask,render\_template

from flask\_mqtt import Mqtt

import ssl

import urllib.request as request

app=Flask(\_\_name\_\_)

app.config['MQTT\_BROKER\_URL']='ailnju7bxk3mm-ats.iot.us-east-1.amazonaws.com'

#app.config['MQTT\_BROKER\_URL']='https://p4tp0s31gh.execute-api.ap-south-1.amazonaws.com'

app.config['MQTT\_BROKER\_PORT']=8883

app.config['MQTT\_CLIENT\_ID']="geocodes"

app.config['MQTT\_KEEPALIVE']=60

app.config['MQTT\_TLS\_ENABLED']=True

app.config['MQTT\_TLS\_CA\_CERTS']="AmazonRootCA1.pem"

app.config['MQTT\_TLS\_CERTFILE']="aece684b8e-certificate.pem.crt"

app.config['MQTT\_TLS\_KEYFILE']="aece684b8e-private.pem.key"

app.config['MQTT\_TLS\_CIPHERS']=None

app.config['MQTT\_TLS\_CERT\_REQS']=ssl.CERT\_REQUIRED

app.config['MQTT\_TLS\_VERSION']=ssl.PROTOCOL\_TLSv1\_2

mqtt=Mqtt(app)

lat=12.9716

lng=77.5946

api\_key='Y7dIJIrCFqMCrLzxlJ1HnEsa76TDkQvG619MzrRxspc'

@mqtt.on\_log()

def handle\_logging(client, userdata, level, buf):

print(level, buf)

#api\_key= 'waD6G64LINl7ieNoERaJ'

#@mqtt.on\_log()

#def handle\_logging(client, userdata, level, buf):

# if level == MQTT\_LOG\_ERR:

# print('Error: {}'.format(buf))

@mqtt.on\_connect()

def handle\_connect(client,userdata,flags,rc):

mqtt.subscribe('iot')

# if rc==0:

# print("connected OK Returned code=",rc)

# else:

# print("Bad connection Returned code=",rc)

# print('Connected and waiting for msgs')

mqtt.publish('home/mytopic', 'hello world')

@mqtt.on\_message()

def handle\_mqtt\_messages(client,userdata,msg):

# print("Msg "+msg)

global lat,lng

data=eval(msg.payload.decode())

lat=data['latitude']

lng=data['longitude']

print("received from aws Lat="+str(lat)+" Long"+str(lng))

print("MQTT Setup done "+"initial value -Lat="+str(lat)+" long="+str(lng))

#Flask code

#app = Flask(\_\_name\_\_)

@app.route('/')

def map\_func():

print("before render template apikey="+api\_key)

return render\_template('map.html',apikey=api\_key,latitude=lat, longitude=lng) #map.html is my HTML file name

# return "<h1>Distant Reading Archive</h1><p>This site is a prototype API for distant reading of science fiction novels.</p>"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug = False)

# @app.route('/')

# def index():

# return render\_template('home.html')

# @app.route('/getlatlng')

# def getlatlng\_page():

# global lat,lng

# print("called ajax:",str(lat)+','+str(lng))

# return str(lat)+','+str(lng)

# if \_\_name\_\_=='\_\_main\_\_':

# app.run(use\_reloader=False,debug=True)

**Device :-**

import json

import paho.mqtt.client as paho

import os

import socket

import ssl

from time import sleep

import random

from random import uniform

connflag = False

#print(ssl.OPENSSL\_VERSION)

def on\_connect(client,userdata,flags,rc):

global connflag

connflag =True

print("Connection required result:"+str(rc))

def on\_message(client,userdata,msg):

print(msg.topic+" "+str(msg.payload))

mqttc = paho.Client()

mqttc.on\_connect = on\_connect

mqttc.on\_message = on\_message

latitude = 12.9716

longitude = 77.5946

awshost = "ailnju7bxk3mm-ats.iot.us-east-1.amazonaws.com"

awsport = 8883

clientId = "geocode"

thingName="geocode"

caPath = "AmazonRootCA1.pem"

certPath = "f3fc725581-certificate.pem.crt"

keyPath = "f3fc725581-private.pem.key"

mqttc.tls\_set(caPath,certfile=certPath,keyfile=keyPath,cert\_reqs=ssl.CERT\_REQUIRED,tls\_version=ssl.PROTOCOL\_TLSv1\_2,ciphers=None)

mqttc.connect(awshost,awsport,keepalive=60)

mqttc.loop\_start()

while 1==1:

sleep(5)

if connflag == True:

new\_lat = latitude+random.random()/100

new\_lon = longitude+random.random()/100

payload = json.dumps({'latitude':new\_lat,'longitude':new\_lon})

mqttc.publish("iot", payload, qos=0)

print("msg sent: " + "%s"%payload )

else:

print("waiting for connection...")

**Html Code :-**

<html>

<head>

<meta name="viewport" content="initial-scale=1.0, width=device-width" />

<script src="https://js.api.here.com/v3/3.1/mapsjs-core.js"type="text/javascript" charset="utf-8"></script>

<script src="https://js.api.here.com/v3/3.1/mapsjs-service.js"type="text/javascript" charset="utf-8"></script>

<script type = "text/JavaScript">

<!--

function AutoRefresh(t) {

setTimeout("location.reload(true);", t);

}

//-->

</script>

</head>

<body style='margin: 0' onload="JavaScript:AutoRefresh(20000);">

<div style="width: 100vw; height: 100vh" id="mapContainer"></div>

<script>

// Initialize the platform object:

var platform = new H.service.Platform({

'apikey': '{{apikey}}'

});

const lng = {{longitude}};

const lat = {{latitude}};

// Obtain the default map types from the platform object

var defaultLayers = platform.createDefaultLayers();

// Instantiate (and display) a map object:

var map = new H.Map(

document.getElementById('mapContainer'),

defaultLayers.vector.normal.map,

{

zoom: 10,

center: { lat: lat, lng: lng }

});

const marker = new H.map.Marker({lat: lat, lng: lng});

map.addObject(marker);

console.log(lng);

</script>

</body>

</html>