TEAM I’d –PNT2022TMID21920

FINAL CODE

# PYTHON CODE:

import wiotp.sdk.device import time import json myConfig = { "identity": {

"orgId": "crmwpw", "typeId": "childdevice", "deviceId":"CHILD"

},

"auth": {

"token": "1234567890"

}

}

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None) client.connect() while True:

name="smartbridge" #in area location latitude=11.651145 longitude=78.156674 #out area location #latitude=11.651165 #longitude=78.158672

myData={'name':name, 'lat':latitude, 'lon':longitude} client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

print("Published data Successfully: %s", myData) time.sleep(5) client.disconnect()

# ADDING GEOFENCE:

package com.example.geofence; import android.app.PendingIntent; import android.content.Context; import android.content.ContextWrapper; import android.content.Intent; import android.widget.Toast; import com.google.android.gms.common.api.ApiException; import com.google.android.gms.location.Geofence; import com.google.android.gms.location.GeofenceStatusCodes; import com.google.android.gms.location.GeofencingRequest; import com.google.android.gms.maps.model.LatLng; public class GeofenceHelper extends ContextWrapper

{

private static final String TAG = "GeofenceHelper"; PendingIntent pendingIntent; public GeofenceHelper(Context base)

{ super(base);

}

public GeofencingRequest getGeofencingRequest(Geofence geofence)

{

return new GeofencingRequest.Builder()

.addGeofence(geofence)

.setInitialTrigger(GeofencingRequest.INITIAL\_TRIGGER\_ENTER)

.build(); }

public Geofence getGeofence(String ID, LatLng latLng, float radius, int transitionTypes)

{

return new Geofence.Builder()

.setCircularRegion(latLng.latitude, latLng.longitude, radius)

.setRequestId(ID)

.setTransitionTypes(transitionTypes)

.setLoiteringDelay(5000)

.setExpirationDuration(Geofence.NEVER\_EXPIRE)

.build();

}

public PendingIntent getPendingIntent()

{

if (pendingIntent != null)

{

return pendingIntent;

}

Intent intent = new Intent(this, GeofenceBroadcastReceiver.class); pendingIntent = PendingIntent.getBroadcast(this, 2607, intent, PendingIntent.FLAG\_IMMUTABLE);

return pendingIntent;

}

public String getErrorString(Exception e)

{

if (e instanceof ApiException)

{

ApiException apiException = (ApiException) e; switch (apiException.getStatusCode())

{ case GeofenceStatusCodes

.GEOFENCE\_NOT\_AVAILABLE:

return "GEOFENCE\_NOT\_AVAILABLE"; case GeofenceStatusCodes

.GEOFENCE\_TOO\_MANY\_GEOFENCES:

return "GEOFENCE\_TOO\_MANY\_GEOFENCES";

case GeofenceStatusCodes

.GEOFENCE\_TOO\_MANY\_PENDING\_INTENTS:

return "GEOFENCE\_TOO\_MANY\_PENDING\_INTENTS";

} }

return e.getLocalizedMessage();

# ALERT NOTIFICATION:

package com.example.geofence; import android.content.BroadcastReceiver; import android.content.Context; import android.content.Intent; import android.location.Location; import android.os.CountDownTimer;

import android.util.Log; import android.widget.Toast; import com.google.android.gms.location.Geofence; import

com.google.android.gms.location.GeofencingEvent import java.util.List; import android.os.Handler;

public class GeofenceBroadcastReceiver extends BroadcastReceiver

{

private static final String TAG = "GeofenceBroadcastReceiv"; @Override public void onReceive(Context context, Intent intent)

{

// TODO: This method is called when the BroadcastReceiver is receiving // an Intent broadcast

//. /\*

Toast.makeText(context, "GEOFENCE\_ENTERED", Toast.LENGTH\_SHORT).show(); final Toast mToastToShow; int toastDurationInMilliSeconds = 1200000; mToastToShow

= Toast.makeText(context, "GEOFENCE\_EXITED", Toast.LENGTH\_LONG);

// Set the countdown to display the toast CountDownTimer toastCountDown; toastCountDown = new CountDownTimer(toastDurationInMilliSeconds, 100000)

{

public void onTick(long millisUntilFinished)

{

mToastToShow.show();

}

public void onFinish()

{

mToastToShow.cancel();

}

}; // Show the toast and starts the countdown mToastToShow.show(); toastCountDown.start();\*/

NotificationHelper notificationHelper = new NotificationHelper(context); notificationHelper.sendHighPriorityNotification("GEOFENCE\_TRANSITION\_ENT ER", "",

MapsActivity.class);

GeofencingEvent geofencingEvent = GeofencingEvent.fromIntent(intent); if (geofencingEvent.hasError())

Log.d(TAG, "onReceive: Error receiving geofence event..."); return;

}

List geofenceList = geofencingEvent.getTriggeringGeofences (); for (Geofence geofence: geofenceList)

{

Log.d(TAG, "onReceive: " + geofence.getRequestId());

}

// Location location = geofencingEvent.getTriggeringLocation(); int transitionType = geofencingEvent.getGeofenceTransition(); switch (transitionType)

{

case Geofence.GEOFENCE\_TRANSITION\_ENTER: notificationHelper.sendHighPriorityNotification("Entered the Location", "", MapsActivity.class); break; case Geofence.GEOFENCE\_TRANSITION\_EXIT:

notificationHelper.sendHighPriorityNotification("Exited the Location ", "", MapsActivity.class);

break; } }

}