package com.example.location;

import android.content.IntentSender;

import android.location.Location;

import android.os.Bundle;

import android.support.v4.app.FragmentActivity;

import android.view.Menu;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.GooglePlayServicesClient;

import com.google.android.gms.location.LocationClient;

import com.google.android.gms.location.LocationRequest;

import com.google.android.gms.location.LocationListener;

public class MainActivity extends FragmentActivity implements GooglePlayServicesClient.ConnectionCallbacks,

GooglePlayServicesClient.OnConnectionFailedListener,

LocationListener{

private final static int

CONNECTION\_FAILURE\_RESOLUTION\_REQUEST = 9000;

LocationClient client;

TextView tv1;

Button bt;

TextView updates;

Button getUpdates;

Button stopUpdates;

// Milliseconds per second

private static final int MILLISECONDS\_PER\_SECOND = 1000;

// Update frequency in seconds

public static final int UPDATE\_INTERVAL\_IN\_SECONDS = 5\*60;

// Update frequency in milliseconds

private static final long UPDATE\_INTERVAL =

MILLISECONDS\_PER\_SECOND \* UPDATE\_INTERVAL\_IN\_SECONDS;

// The fastest update frequency, in seconds

private static final int FASTEST\_INTERVAL\_IN\_SECONDS = 1\*60;

// A fast frequency ceiling in milliseconds

private static final long FASTEST\_INTERVAL =

MILLISECONDS\_PER\_SECOND \* FASTEST\_INTERVAL\_IN\_SECONDS;

LocationRequest mLocationRequest;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

tv1 = (TextView)findViewById(R.id.tv1);

bt = (Button)findViewById(R.id.bt);

updates =(TextView)findViewById(R.id.updates);

getUpdates =(Button)findViewById(R.id.getUpdate);

stopUpdates=(Button)findViewById(R.id.stopUpdate);

client = new LocationClient(this, this, this);

bt.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

getLocation();

}

});

mLocationRequest = LocationRequest.create();

mLocationRequest.setPriority(LocationRequest.PRIORITY\_HIGH\_ACCURACY);

mLocationRequest.setInterval(UPDATE\_INTERVAL);

mLocationRequest.setFastestInterval(FASTEST\_INTERVAL);

getUpdates.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

startPeriodicUpdates();

}

});

stopUpdates.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

stopPeriodicUpdates();

}

});

}

private void getLocation(){

Location loc;

loc = client.getLastLocation();

tv1.setText(locationStringFromLocation(loc));

}

private String locationStringFromLocation(final Location location) {

return Location.convert(location.getLatitude(), Location.FORMAT\_DEGREES) + " " + Location.convert(location.getLongitude(), Location.FORMAT\_DEGREES);

}

private void startPeriodicUpdates() {

client.requestLocationUpdates(mLocationRequest, (LocationListener)this);

updates.setText(R.string.location\_requested);

}

private void stopPeriodicUpdates() {

client.removeLocationUpdates(this);

updates.setText(R.string.location\_updates\_stopped);

}

@Override

protected void onStart() {

super.onStart();

// Connect the client.

client.connect();

}

@Override

protected void onStop() {

if (client.isConnected()) {

stopPeriodicUpdates();

}

// Disconnecting the client invalidates it.

client.disconnect();

super.onStop();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

@Override

public void onConnectionFailed(ConnectionResult result) {

// TODO Auto-generated method stub

if (result.hasResolution()) {

try {

// Start an Activity that tries to resolve the error

result.startResolutionForResult(

this,

CONNECTION\_FAILURE\_RESOLUTION\_REQUEST);

/\*

\* Thrown if Google Play services canceled the original

\* PendingIntent

\*/

} catch (IntentSender.SendIntentException e) {

// Log the error

e.printStackTrace();

}

} else {

/\*

\* If no resolution is available, display a dialog to the

\* user with the error.

\*/

Toast.makeText(MainActivity.this,result.getErrorCode() , Toast.LENGTH\_SHORT).show();

}

}

@Override

public void onConnected(Bundle connectionHint) {

// TODO Auto-generated method stub

Toast.makeText(MainActivity.this, "Connected", Toast.LENGTH\_LONG).show();

}

@Override

public void onDisconnected() {

// TODO Auto-generated method stub

Toast.makeText(MainActivity.this, "Dis-Connected", Toast.LENGTH\_LONG).show();

}

@Override

public void onLocationChanged(Location location) {

// TODO Auto-generated method stub

// Report to the UI that the location was updated

String msg = "Updated Location: " +

Double.toString(location.getLatitude()) + "," +

Double.toString(location.getLongitude());

Toast.makeText(this, msg, Toast.LENGTH\_SHORT).show();

}

}