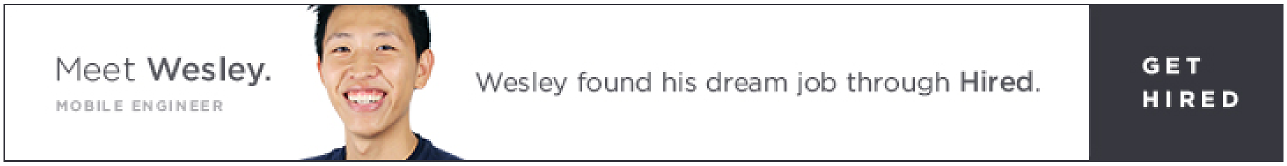
ANDROIDHIVE



Android GPS, Location Manager Tutorial

0 comments . By Ravi Tamada . on July 1, 2012 Tweet

If you are developing any location based app, you can make your app more smarter by finding user’s location automatically. For this you need to integrate location modules in your application. This tutorial covers how to integrate location module in your apps.

DOWNLOAD CODE

Note:

This tutorial uses older APIs to fetch user's location. Now google introduced new way of getting user's location using Google Play Services. Read Android Location API using Google Play Services to get the location using play services.

Creating new Android Project

1. Create a new project in Eclipse by navigating to **File ⇒ New ⇒ Android Project** and fill all the required details.

2. Open AndroidManifest.xml and add `ACCESS_FINE_LOCATION` (Which includes both ACCESS_FINE_LOCATION and ACCESS_COARSE_LOCATION). Also if you are getting network-based location then you need to add `INTERNET` permission too.

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.gpstracking"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="8" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".AndroidGPSTrackingActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />

</manifest>
```

Writing GPS Manager Class

3. Create a new class and name it as **GPSTracker.java** and extend the calls from **Service**. Also implement this class from **LocationListener**.

```
public class GPSTracker extends Service implements LocationListener{
```

4. Add the required global variables and a constructor for this class.

```
public class GPSTracker extends Service implements LocationListener {

    private final Context mContext;

    // flag for GPS status
    boolean isGPSEnabled = false;

    // flag for network status
    boolean isNetworkEnabled = false;

    boolean canGetLocation = false;

    Location location; // location
    double latitude; // latitude
    double longitude; // longitude

    // The minimum distance to change Updates in meters
    private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10; // 10 meters

    // The minimum time between updates in milliseconds
    private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1 minute

    // Declaring a Location Manager
    protected LocationManager locationManager;

    public GPSTracker(Context context) {
        this.mContext = context;
        getLocation();
    }
```

5. Add the implementation for the function **geoLocation()** which was called in the constructor. In this function we'll get the location from network provider first. If network provider is disabled, then we get the location from GPS provider.

GPSTRACKER.JAVA

```
public Location getLocation() {
        try {
            locationManager = (LocationManager) mContext
                    .getSystemService(LOCATION_SERVICE);

            // getting GPS status
            isGPSEnabled = locationManager
                    .isProviderEnabled(LocationManager.GPS_PROVIDER);

            // getting network status
            isNetworkEnabled = locationManager
                    .isProviderEnabled(LocationManager.NETWORK_PROVIDER);

            if (!isGPSEnabled && !isNetworkEnabled) {
                // no network provider is enabled
            } else {
                this.canGetLocation = true;
                // First get location from Network Provider
                if (isNetworkEnabled) {
                    locationManager.requestLocationUpdates(
                            LocationManager.NETWORK_PROVIDER,
                            MIN_TIME_BW_UPDATES,
                            MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                    Log.d("Network", "Network");
                    if (locationManager != null) {
                        location = locationManager
                                .getLastKnownLocation(LocationManager.NETWORK_PR
                        if (location != null) {
                            latitude = location.getLatitude();
                            longitude = location.getLongitude();
                        }
                    }
                }
                // if GPS Enabled get lat/long using GPS Services
                if (isGPSEnabled) {
                    if (location == null) {
                        locationManager.requestLocationUpdates(
                                LocationManager.GPS_PROVIDER,
                                MIN_TIME_BW_UPDATES,
                                MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                        Log.d("GPS Enabled", "GPS Enabled");
                        if (locationManager != null) {
                            location = locationManager
                                    .getLastKnownLocation(LocationManager.GPS_PR
                            if (location != null) {
                                latitude = location.getLatitude();
                                longitude = location.getLongitude();
                            }
                        }
                    }
                }
            }

        } catch (Exception e) {
            e.printStackTrace();
        }

        return location;
    }
    @Override
    public void onLocationChanged(Location location) {
    }

    @Override
    public void onProviderDisabled(String provider) {
    }

    @Override
    public void onProviderEnabled(String provider) {
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
    }

    @Override
    public IBinder onBind(Intent arg0) {
        return null;
    }
```

Getting user’s current location (Latitude and Longitude)

6. Add the following functions to **GPSTracker.java**. (These functions will return 0.00 if it fails to get latitude and longitude)

GPSTRACKER.JAVA

```
/**
     * Function to get latitude
     * */
    public double getLatitude(){
        if(location != null){
            latitude = location.getLatitude();
        }

        // return latitude
        return latitude;
    }

    /**
     * Function to get longitude
     * */
    public double getLongitude(){
        if(location != null){
            longitude = location.getLongitude();
        }

        // return longitude
        return longitude;
    }
```

Prompting users to Turn On GPS

7. If user turned off the GPS we can use ask user to enable GPS. The following code will show an Alert message asking user to turn on GPS by navigating to GPS Settings automatically.

GPSTRACKER.JAVA

```
/**
     * Function to check if best network provider
     * @return boolean
     * */
    public boolean canGetLocation() {
        return this.canGetLocation;
    }

    /**
     * Function to show settings alert dialog
     * */
    public void showSettingsAlert(){
        AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);

        // Setting Dialog Title
        alertDialog.setTitle("GPS is settings");

        // Setting Dialog Message
        alertDialog.setMessage("GPS is not enabled. Do you want to go to setting
        // Setting Icon to Dialog
        //alertDialog.setIcon(R.drawable.delete);

        // On pressing Settings button
        alertDialog.setPositiveButton("Settings", new DialogInterface.OnClickLis
            public void onClick(DialogInterface dialog,int which) {
                Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTI
                mContext.startActivity(intent);
            }
        });

        // on pressing cancel button
        alertDialog.setNegativeButton("Cancel", new DialogInterface.OnClickListe
            public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
            }
        });

        // Showing Alert Message
        alertDialog.show();
    }
```

Stopping the Location Service

Calling following function will stop using location updates in your application.

```
/**
     * Stop using GPS listener
     * Calling this function will stop using GPS in your app
     * */
    public void stopUsingGPS(){
        if(locationManager != null){
            locationManager.removeUpdates(GPSTracker.this);
        }
    }
```

Final Code

Below is the complete code of GPSTracker.java

GPSTracker.java

```
package com.example.gpstracking;

import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.provider.Settings;
import android.util.Log;

public class GPSTracker extends Service implements LocationListener {

    private final Context mContext;

    // flag for GPS status
    boolean isGPSEnabled = false;

    // flag for network status
    boolean isNetworkEnabled = false;

    // flag for GPS status
    boolean canGetLocation = false;

    Location location; // location
    double latitude; // latitude
    double longitude; // longitude

    // The minimum distance to change Updates in meters
    private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10; // 10 meters

    // The minimum time between updates in milliseconds
    private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1 minute

    // Declaring a Location Manager
    protected LocationManager locationManager;

    public GPSTracker(Context context) {
        this.mContext = context;
        getLocation();
    }
```

```
public Location getLocation() {
    try {
        locationManager = (LocationManager) mContext
                .getSystemService(LOCATION_SERVICE);

        // getting GPS status
        isGPSEnabled = locationManager
                .isProviderEnabled(LocationManager.GPS_PROVIDER);

        // getting network status
        isNetworkEnabled = locationManager
                .isProviderEnabled(LocationManager.NETWORK_PROVIDER);

        if (!isGPSEnabled && !isNetworkEnabled) {
            // no network provider is enabled
        } else {
            this.canGetLocation = true;
            // First get location from Network Provider
            if (isNetworkEnabled) {
                locationManager.requestLocationUpdates(
                        LocationManager.NETWORK_PROVIDER,
                        MIN_TIME_BW_UPDATES,
                        MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                Log.d("Network", "Network");
                if (locationManager != null) {
                    location = locationManager
                            .getLastKnownLocation(LocationManager.NETWORK_PR
                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
            // if GPS Enabled get lat/long using GPS Services
            if (isGPSEnabled) {
                if (location == null) {
                    locationManager.requestLocationUpdates(
                            LocationManager.GPS_PROVIDER,
                            MIN_TIME_BW_UPDATES,
                            MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                    Log.d("GPS Enabled", "GPS Enabled");
                    if (locationManager != null) {
                        location = locationManager
                                .getLastKnownLocation(LocationManager.GPS_PR
                        if (location != null) {
                            latitude = location.getLatitude();
                            longitude = location.getLongitude();
                        }
                    }
                }
            }
        }

    } catch (Exception e) {
        e.printStackTrace();
    }
```

```
        return location;
    }

    /**
     * Stop using GPS listener
     * Calling this function will stop using GPS in your app
     * */
    public void stopUsingGPS(){
        if(locationManager != null){
            locationManager.removeUpdates(GPSTracker.this);
        }
    }

    /**
     * Function to get latitude
     * */
    public double getLatitude(){
        if(location != null){
            latitude = location.getLatitude();
        }

        // return latitude
        return latitude;
    }

    /**
     * Function to get longitude
     * */
    public double getLongitude(){
        if(location != null){
            longitude = location.getLongitude();
        }

        // return longitude
        return longitude;
    }

    /**
     * Function to check GPS/wifi enabled
     * @return boolean
     * */
    public boolean canGetLocation() {
        return this.canGetLocation;
    }

    /**
     * Function to show settings alert dialog
     * On pressing Settings button will lauch Settings Options
     * */
    public void showSettingsAlert(){
        AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);

        // Setting Dialog Title
        alertDialog.setTitle("GPS is settings");

        // Setting Dialog Message
        alertDialog.setMessage("GPS is not enabled. Do you want to go to setting
        // On pressing Settings button
        alertDialog.setPositiveButton("Settings", new DialogInterface.OnClickLis
            public void onClick(DialogInterface dialog,int which) {
                Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTI
                mContext.startActivity(intent);
            }
        });

        // on pressing cancel button
        alertDialog.setNegativeButton("Cancel", new DialogInterface.OnClickListe
            public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
            }
        });

        // Showing Alert Message
        alertDialog.show();
    }

    @Override
    public void onLocationChanged(Location location) {
    }

    @Override
    public void onProviderDisabled(String provider) {
    }

    @Override
    public void onProviderEnabled(String provider) {
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
    }

    @Override
    public IBinder onBind(Intent arg0) {
        return null;
    }

}
```

How to Use

8. You can get user’s current location by calling simple function from GPSTracker class. Open your main activity and try the following code.

Check gps enabled or not

```
GPSTracker gps = new GPSTracker(this);
if(gps.canGetLocation()){ // gps enabled} // return boolean true/false
```

Getting Latitude and Longitude

```
gps.getLatitude(); // returns latitude
gps.getLongitude(); // returns longitude
```

Showing GPS Settings Alert Dialog

```
gps.showSettingsAlert();
```

Stop using GPS

```
gps.stopUsingGPS();
```

ANDROIDGPSTRACKINGACTIVITY.JAVA

```
package com.example.gpstracking;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class AndroidGPSTrackingActivity extends Activity {

    Button btnShowLocation;

    // GPSTracker class
    GPSTracker gps;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

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

        // show location button click event
        btnShowLocation.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // create class object
                gps = new GPSTracker(AndroidGPSTrackingActivity.this);

                // check if GPS enabled
                if(gps.canGetLocation()){

                    double latitude = gps.getLatitude();
                    double longitude = gps.getLongitude();

                    // \n is for new line
                    Toast.makeText(getApplicationContext(), "Your Location is - \
                }else{
                    // can't get location
                    // GPS or Network is not enabled
                    // Ask user to enable GPS/network in settings
                    gps.showSettingsAlert();
                }

            }
        });
    }

}
```

Testing your GPS App in Emulator using DDMS Tool

You can test your application in different ways. If you have real device you can directly test the application by installing your application. If you don’t have one, you can test the app using local emulator.

After starting the emulator open DDMS tool form **EClipse Windows** ⇒ **Show Perspective** ⇒ **DDMS** (Also you can find it on the right corner of IDE)

In the DDMS tool you can find list of emulators you opened. Select the appropriate emulator. In Emulator Controls tab you can manually pass your latitude and longitude to emulator.

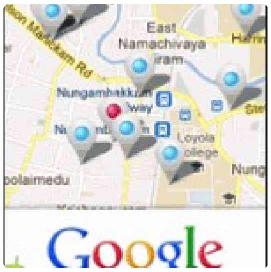
You can find Emulator testing in demo video of this tutorial.

Share this article on

Tweet

ANDROIDHIVE

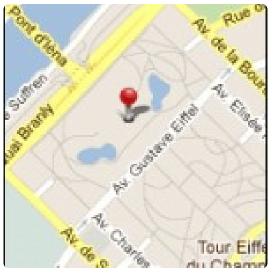
You May Also Like



Android Working with Google Places and Maps Tutorial

Android working with Google Maps V2

Android Location API using Google Play Services



Android Working with Google Maps

Ravi Tamada
Hyderabad, INDIA

Subscribe to get latest updates to your inbox.
-- I don’t spam!

Enter your email here SUBSCRIBE

ANDROIDHIVE

Tag Cloud

ANDROIDHIVE

Action Bar Adapter Analytics
Animation API Apps Async
Beginner Camera Chat Dashboard
Database facebook File Upload
Fragments GCM GDK Gestures
Google Google Glass Google Plus
GPS Grid Grid View HTTP
Intermediate json Libstreaming
List View Locale Location Maps
Material Design MySQL
Navigation Drawer PayPal PHP Pinch
Progress Bar Push Notifications
Quick Tips RecyclerView REST
sessions Slim SMS Sockets
Speech Input Spinner sponsored
SQLite Swipe Tab View Twitter UI
Video Video Streaming View Pager
Volley Wearable xml YouTube

Most Popular

1 Android SQLite Database Tutorial - 1,240,709 views

2 How to connect Android with PHP, MySQL - 1,193,304 views

3 Android JSON Parsing Tutorial - 1,073,248 views

4 Android Push Notifications using Google Cloud Messaging (GCM), PHP and MySQL - 989,033 views

ANDROIDHIVE

5 Android Custom ListView with Image and Text - 888,351 views

6 Android Sliding Menu using Navigation Drawer - 882,540 views

7 Android Login and Registration with PHP, MySQL and SQLite - 785,473 views

8 Android GPS, Location Manager Tutorial - 598,160 views

9 Android Tab Layout with Swipeable Views - 576,057 views

10 Android Tab Layout Tutorial - 548,838 views