ANDROID LOCATION API USING GOOGLE PLAY SERVICES

In this tutorial, we are going to learn how to implement android location service api using Google Play Services.

The previous android tutorial I wrote on this topic was on ***Android Framework Location API using Android LocationManager***. As I stated in the tutorial that Google has suggested that to best way to work with android Location API is to use the Google Play Services.

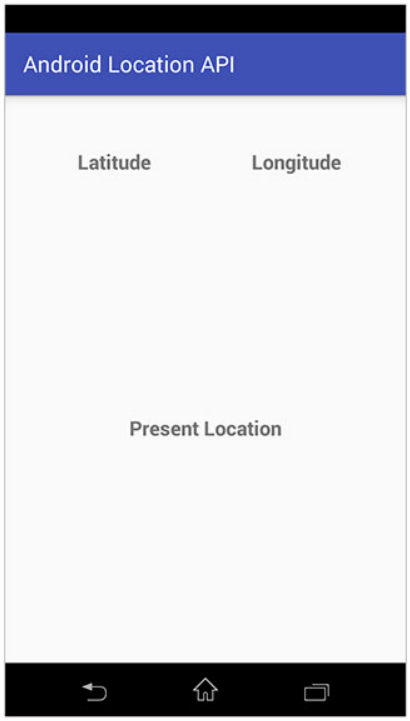
This is what we are going to use in this tutorial. Unlike this previous tutorial, here where are going to add the Google Play Services to our application.

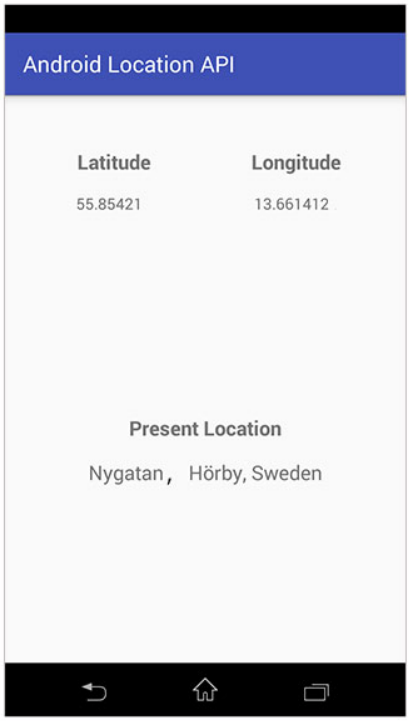
Android Google Play Services helps you take advantage of the latest,

If you want to read more about Google Play Service, you can refer to the link in ***android developers guide***.

Note: If the number of method references in your app exceeds the 65K limit, your app may fail to compile. You may be able to mitigate this problem when compiling your app by specifying only the specific Google Play services APIs your app uses, instead of all of them. For information on how to do this, see Selectively compiling APIs into your executable.

Before we go deeper into this tutorial, it is important for us to understand what we are planning to achieve. Below is the screen-shot of the application we will be creating.





Lets start to soil our hands in code. we will create our project in our IDE. For this tutorial, I am using the following tools and environment, feel free to use what works for you.

Windows 7

Android Studio

To create a new android application project, follow the steps as stipulated below.

Go to File menu

Click on New menu

Click on Android Application

Enter Project name: AndroidLocationServices

Package: com.inducesmile.androidlocationservices

Select Blank Activity

Keep other default selections

Continue to click on next button until Finish button is active, then click on Finish Button.

Now that we have created our android project, we are going to add Google Play Services as one of our application dependency module.

Open your android build.gradle file located in gradle script/build.gradle , copy and paste this line of code to the dependencies section.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.google.android.gms:play-services:8.4.0'
}
```

Go to the res folder, then double click on the values folder and double click on the colors.xml file and add the following lines of codes. This will holder all the colors we will use in this tutorial.

- Android Retrofit 2 with JSON API Example
- Android Room with LiveData Example Tutorial
- Android UI App for Food Ordering and Delivery from Multiple Restaurants

```
    <color name="colorBlack">#000000</color>
</resources>
```

Now, head over to the strings.xml file and modify the content with the code below.

```
<resources>
    <string name="app_name">Android Google Play Services
    <string name="latitude">Latitude</string>
    <string name="longitude">Longitude</string>
    <string name="city">Present Location</string>
    <string name="no_text"> </string>
</resources>
```

Since we are going to use Location API, we will use permission to access the location. The user must grant the request before the app can access the user location.

Open you Mainfest.xml , copy and paste the following code inside the file. The complete code for the Mainfest.xml file will look like this.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/
    package="com.inducesmile.androidlocationservices">

    <uses-permission android:name="android.permission.AC
    <uses-permission android:name="android.permission.AC
    <uses-permission android:name="android.permission.IN

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.act
```

```
</manifest>
```

The main layout file for the MainActivity class is located in the res/layout folder. Double click on activity_main.xml file to open it in your IDE.

We are going to add six TextView widgets. Three of them will holder the title information while the remaining three will be set when we access the longitude and latitude values of users location. We will further use the Geocoder class to obtain the address of the user.

Copy and paste the below code in activity_main.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_marg
    android:paddingLeft="@dimen/activity_horizontal_marg
    android:paddingRight="@dimen/activity_horizontal_mar
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="horizontal">
        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="vertical">
            <TextView
                android:id="@+id/latitude_title"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/latitude"
                android:textSize="18dp"
```

```
            android:id="@+id/latitude"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/no_text"
            android:textColor="@color/colorBlack"
            android:layout_marginTop="16dp"
            android:layout_gravity="center"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical">
        <TextView
            android:id="@+id/longitude_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/longitude"
            android:textSize="18dp"
            android:textStyle="bold"
            android:textColor="@color/colorBlack"
            android:layout_marginTop="32dp"
            android:layout_gravity="center"/>
        <TextView
            android:id="@+id/longitude"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/no_text"
            android:textColor="@color/colorBlack"
            android:layout_marginTop="16dp"
            android:layout_gravity="center"/>
    </LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="vertical">
    <TextView
        android:id="@+id/city_location"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
            android:layout_marginTop="32dp"
            android:layout_gravity="center_horizontal"/>
        <TextView
            android:id="@+id/city"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/no_text"
            android:textSize="18dp"
            android:textColor="@color/colorBlack"
            android:layout_marginTop="16dp"
            android:layout_gravity="center_horizontal"/>
    </LinearLayout>

</LinearLayout>
```

As you can see, the layout file is very simple and straight forward. We will now move over to the MainActivity class.

We will going to obtain the instances of the TextView controls that we will populate with latitude and longitude.

In the Mainactivity class, we are going to implement the following interfaces. Add the following line in the class declaration

```
public class MainActivity extends AppCompatActivity impl
```

We add the GoogleAPiClient as a class variable

```
private GoogleApiClient mGoogleApiClient;
```

Instanciate the GoogleAPiClient in the onCreate() method of the Activity class

```
if (mGoogleApiClient == null) {
    mGoogleApiClient = new GoogleApiClient.Builder(this)
            .addConnectionCallbacks(this)
            .addOnConnectionFailedListener(this)
```

In the onStart() method, connect the GoogleApiClient instance to Google Play Services as shown below

```
@Override
protected void onStart() {
    mGoogleApiClient.connect();
    super.onStart();
}
```

In onStop() method, the GoogleApiClient is disconnected in order to free resources when the activity is finally destroyed.

```
@Override
protected void onStop() {
    mGoogleApiClient.disconnect();
    super.onStop();
}
```

In the onConnected(Bundle bundle) callback method which signified that the GoogleApiClient instance is connected, LocationRequest is send to see if there is any settings that is not enable that the Location service Api might need.

The setResultCallback() method of the result instance checks if all the setting required is enable, if so the last known location is requested from from the FusedLocationApi class .

Please not that starting from Android Api level 23, that permissions are requested at run-time so we have to always request user permission. The down side of it is that if a user refuse to grant a permission that our application will use, we will find a way to handle it so that it will not crash our application or make it behave in a way that was not intended.

The complete code on this method is shown below.

```
@Override
public void onConnected(Bundle bundle) {
    mLocationRequest = createLocationRequest();
```

```
        public void onResult(LocationSettingsResult resu
            final Status status = result.getStatus();
            final LocationSettingsStates mState = result
            switch (status.getStatusCode()) {
                case LocationSettingsStatusCodes.SUCCESS
                    // All location settings are satisfi
                    // initialize location requests here
                    if (ActivityCompat.checkSelfPermissi
                        ActivityCompat.requestPermission
                    } else {
                        mLastLocation = LocationServices
                        if (mLastLocation != null) {
                            latitudePosition.setText(Str
                            longitudePosition.setText(St
                            getAddressFromLocation(mLast
                        }
                    }
                    break;
                case LocationSettingsStatusCodes.RESOLUT
                    // Location settings are not satisfi
                    // by showing the user a dialog.
                    try {
                        // Show the dialog by calling st
                        // and check the result in onAct
                        status.startResolutionForResult(
                    } catch (IntentSender.SendIntentExce
                        // Ignore the error.
                    }
                    break;
                case LocationSettingsStatusCodes.SETTING
                    // Location settings are not satisfi
                    // to fix the settings so we won't s
                    break;
            }
        }
    });
}
```

When the location settings is not satisfied, the line below.

```
                                                    */
```

Accept

call the onActivityResult() to give the user to option of enabling what is missing. The code snippet is shown below.

```
@Override
protected void onActivityResult(int requestCode, int res
    switch (requestCode) {
        case REQUEST_CHECK_SETTINGS:
            switch (resultCode) {
                case Activity.RESULT_OK:
                    if (ActivityCompat.checkSelfPermissi
                        ActivityCompat.requestPermission
                    } else {
                        mLastLocation = LocationServices
                        if (mLastLocation != null) {
                            latitudePosition.setText(Str
                            longitudePosition.setText(St
                            getAddressFromLocation(mLast
                        }
                    }
                    break;
                case Activity.RESULT_CANCELED:
                    // user does not want to update sett
                    Toast.makeText(MainActivity.this, "U
                    break;
            }
            break;
    }
}
```

The complete complete for the MainActivity class is shown below.

```
import android.Manifest;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.IntentSender;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
```

```
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient
import com.google.android.gms.common.api.PendingResult;
import com.google.android.gms.common.api.ResultCallback;
import com.google.android.gms.common.api.Status;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.LocationSettingsR
import com.google.android.gms.location.LocationSettingsR
import com.google.android.gms.location.LocationSettingsS
import com.google.android.gms.location.LocationSettingsS
import java.io.IOException;
import java.util.List;
import java.util.Locale;
public class MainActivity extends AppCompatActivity impl
    private static final String TAG = "MainActivity";
    private GoogleApiClient mGoogleApiClient;
    private Location mLastLocation;
    private TextView latitudePosition;
    private TextView longitudePosition;
    private TextView currentCity;
    private final int REQUEST_LOCATION = 200;
    private final int REQUEST_CHECK_SETTINGS = 300;
    private LocationRequest mLocationRequest;
    private PendingResult<LocationSettingsResult> result
    private LocationSettingsRequest.Builder builder;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        latitudePosition = (TextView) findViewById(R.id.
        longitudePosition = (TextView) findViewById(R.id
        currentCity = (TextView) findViewById(R.id.city)
        if (mGoogleApiClient == null) {
            mGoogleApiClient = new GoogleApiClient.Build
                    .addConnectionCallbacks(this)
                    .addOnConnectionFailedListener(this)
```

Accept

```
@Override
protected void onStart() {
    mGoogleApiClient.connect();
    super.onStart();
}
@Override
protected void onStop() {
    mGoogleApiClient.disconnect();
    super.onStop();
}
@Override
public void onConnected(Bundle bundle) {
    mLocationRequest = createLocationRequest();
    builder = new LocationSettingsRequest.Builder().
    result = LocationServices.SettingsApi.checkLocat
    result.setResultCallback(new ResultCallback<Loca
        @Override
        public void onResult(LocationSettingsResult
            final Status status = result.getStatus()
            final LocationSettingsStates mState = re
            switch (status.getStatusCode()) {
                case LocationSettingsStatusCodes.SUC
                    // All location settings are sat
                    // initialize location requests
                    if (ActivityCompat.checkSelfPerm
                        ActivityCompat.requestPermis
                    } else {
                        mLastLocation = LocationServ
                        if (mLastLocation != null) {
                            latitudePosition.setText
                            longitudePosition.setTex
                            getAddressFromLocation(m
                        }
                    }
                    break;
                case LocationSettingsStatusCodes.RES
                    // Location settings are not sat
                    // by showing the user a dialog.
                    try {
                        // Show the dialog by callin
                        // and check the result in o
                        status.startResolutionForRes
```

```
                case LocationSettingsStatusCodes.SET
                    // Location settings are not sat
                    // to fix the settings so we won
                    break;
            }
        }
    });
}
@Override
protected void onActivityResult(int requestCode, int
    switch (requestCode) {
        case REQUEST_CHECK_SETTINGS:
            switch (resultCode) {
                case Activity.RESULT_OK:
                    if (ActivityCompat.checkSelfPerm
                        ActivityCompat.requestPermis
                    } else {
                        mLastLocation = LocationServ
                        if (mLastLocation != null) {
                            latitudePosition.setText
                            longitudePosition.setTex
                            getAddressFromLocation(m
                        }
                    }
                    break;
                case Activity.RESULT_CANCELED:
                    // user does not want to update
                    Toast.makeText(MainActivity.this
                    break;
            }
            break;
    }
}
@Override
public void onRequestPermissionsResult(int requestCo
    switch (requestCode) {
        case REQUEST_LOCATION: {
            // If request is cancelled, the result a
            if (grantResults.length > 0 && grantResu
                // permission was granted, yay! Do t
                // contacts-related task you need to
            } else {
```

Accept

```
            }
            // other 'case' lines to check for other
            // permissions this app might request
        }
    }
    @Override
    public void onConnectionSuspended(int i) {
    }
    @Override
    public void onConnectionFailed(ConnectionResult conn
    }
    @Override
    public void onLocationChanged(Location location) {
        if (mLastLocation != null) {
            latitudePosition.setText(String.valueOf(mLas
            longitudePosition.setText(String.valueOf(mLa
            getAddressFromLocation(mLastLocation, getApp
        }
    }
    public static void getAddressFromLocation(final Loca
        Thread thread = new Thread() {
            @Override
            public void run() {
                Geocoder geocoder = new Geocoder(context
                String result = null;
                try {
                    List<Address> list = geocoder.getFro
                    if (list != null && list.size() > 0)
                        Address address = list.get(0);
                        // sending back first address li
                        result = address.getAddressLine(
                    }
                } catch (IOException e) {
                    Log.e(TAG, "Impossible to connect to
                } finally {
                    Message msg = Message.obtain();
                    msg.setTarget(handler);
                    if (result != null) {
                        msg.what = 1;
                        Bundle bundle = new Bundle();
                        bundle.putString("address", resu
                        msg.setData(bundle);
```

Accept

```
            }
        };
        thread.start();
    }
    private class GeoCoderHandler extends Handler {
        @Override
        public void handleMessage(Message message) {
            String result;
            switch (message.what) {
                case 1:
                    Bundle bundle = message.getData();
                    result = bundle.getString("address")
                    break;
                default:
                    result = null;
            }
            currentCity.setText(result);
        }
    }
    protected LocationRequest createLocationRequest() {
        LocationRequest mLocationRequest = new LocationR
        mLocationRequest.setInterval(10000);
        mLocationRequest.setFastestInterval(5000);
        mLocationRequest.setPriority(LocationRequest.PRI
        return mLocationRequest;
    }
}
```

Please note that I did not test this tutorial on device running android 6 but it has been tested from android 5 and below.

Run your application and see for yourself what we have just created.

This brings us to the end of this tutorial, If you find anything confusing kindly contact me with your questions or use the comment box below.

You can download the code for this tutorial below. If you are having hard time downloading the tutorials, kindly contact me.

Remember to subscribe with your email address so that you will be

think about it.

OTHER INTERESTING POSTS:

Android Code Competition Announcement

Android Facebook SDK Login Integration and Facebook User Profile Display

Android Audio Manager and Volume Control Adjustment

Create a beautiful Android Weather App using OpenWeatherMap API and Volley Library part 2

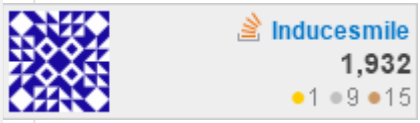
Android Quiz Application with SQLite Database

Android Content Provider - Access SQLite database in one application from another application

ABOUT THE AUTHOR

Inducesmile

I learn and write about Android, iOS, Javascript, Php, Node.js, React Native, Mobile Game, Virtual Reality and Internet of Things



12 COMMENTS

jlively | August 23, 2016 Log in to Reply

Dear Henry,

You always say that the source code can be found below, but I don't see it. Also, where do you use "final LocationSettingsStates mState = result.getLocationSettingsStates();"? It says it has never been used.

Regards,
jay

Henry Author | August 23, 2016
Log in to Reply

The download links is broken in my server so when time permits I will re-upload all the tutorials with there download links.

About your question, you can find what you are looking for here

@Override

Accept

```
final LocationSettingsStates mState =
result.getLocationSettingsStates();
```

jlively | August 23, 2016

Log in to Reply

Dear Henry,

Thank you for the fast response. The problem with the mState is that it says “Variable ‘mState’ is never used”. I can see that you declare it, but where do you use it? I also get the bug that mLastLocation.getLatitude() may produce nullpointer. ‘double android.location.Location.getLatitude()’ on a null object reference

Regards
Jay

Henry Author | August 23, 2016 Log in to Reply

You can check for null point before you use the object if that is the case. I will send the source code to you when time permits.

jlively | August 23, 2016

All right, thank you for your time.

Accept

Dear Henry,
Is there anyway that you can send me the source code?

Behnam | December 9, 2016 Log in to Reply

Thank you man, you saved me..

Henry Author | December 9, 2016
Log in to Reply

I am glad it was helpful to you

Giulio | December 11, 2016 Log in to Reply

Hi Henry,

I ‘ve just looked at your code and tested it and it works, but if I try to move with my smartphone, the coordinates don’t change. Do you know why?

I noticed you put “onLocationChanged” method into the code, but it doesn’t work. Could you please help me?

Henry Author | December 11, 2016
Log in to Reply

The tutorial is to get user location (Latitude and longitude) and convert the Location to its real address using the GeoCoder class. In other to get constant update of you location as you move, you will this method

Accept

https://inducesmile.com/android/android-real-time-user-location-tracking-using-google-map-api-v2/. I hope it will be of help to you.

Giulio | December 11, 2016

Log in to Reply

Hi Henry,

I will look at your tutorial. In the meanwhile I made other tests with this app and if I try to turn the app off and then turn it on, it will properly change coordinates.

Now what I'm looking for is how to integrate this code in a Google Maps API. I will explain better: I got the user location (Latitude and Logitude) and now I want to show this Location in a map using Google Maps API. I tried but it doesn't work.

In the "onCreate" method I have:
SupportMapFragment mapFragment =
(SupportMapFragment)
getSupportFragmentManager()
.findFragmentById(R.id.map);
mapFragment.getMapAsync(this);

and then I have this one:
@Override
public void onMapReady(GoogleMap
googleMap) {
mMap = googleMap;
LatLng currentPosition = new
LatLng(latitudePosition, longitudePosition);
mMap.addMarker(new
MarkerOptions().position(currentPosition).title("Cu
Position"));
mMap.moveCamera(CameraUpdateFactory.newLa

Accept

Thank you in advance!

Giulio | January 17, 2017 Log in to Reply

Dear Henry,
regarding my last question do you have a feedback?
I will explain better: once I got the current location – latitude and longitude – I want to show them on a map, using Google Maps API.
Could you please help me?

ADD A COMMENT

You must be logged in to post a comment.