Inducesmile

Mobile developer community for beginners and students

ANDROID | KOTLIN | IOS (SWIFT) | HYBRID APP ▼ | MOBILE GAMES | IOT – VR

TIPS | CODING HELP | BUY CODE

Home ▸ Android ▸
Android Location API Using Android LocationManager

Search the site

ANDROID LOCATION API USING ANDROID LOCATIONMANAGER

In this android tutorial, we are going to learn to the use android framework location API with LocationManager to get a user last known location.

Location aware android application is among the common features you will see in apps published in Google Play Store.

Because of the mobile nature of hand-held devices and the ability to move around with mobile phones, most android applications will request for user location in order to achieve certain result.

If you want to know where a friend is right now, you can access his or her location through an app. In some cases, it is a vital tool to location where you are going or where you want to be.

ANDROID SOURCE CODE POLL

Which of the below listed source codes should we publish next month?

Multi Restaurant Food Ordering App (27%, 39 Votes)

Stock and Inventory App (18%, 26 Votes)

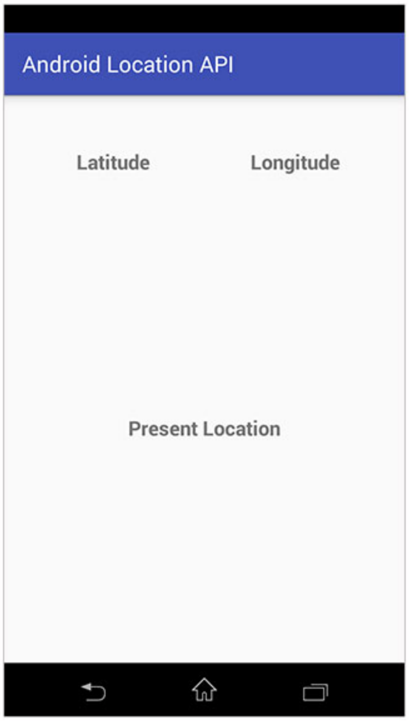
Android Dating App (16%, 23

Accept

According to android developers guide – The Google Play services location APIs are preferred over the Android framework location APIs (android.location) as a way of adding location awareness to your app. If you are currently using the Android framework location APIs, you are strongly encouraged to switch to the Google Play services location APIs as soon as possible.

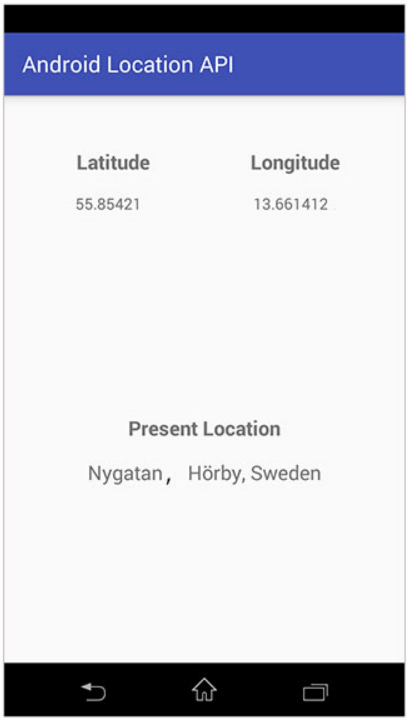
If you want to learn how to implement ***Android Location using Google Play Services***, I will suggest that you click the link and read the tutorials.

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.


Android Location API

Latitude Longitude

Present Location


Android Location API

Latitude Longitude
55.85421 13.661412

Present Location
Nygatan, Hörby, Sweden

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

Android E-Book App (12%, 17 Votes)

Appointment Booking App (10%, 15 Votes)

Business & Service Provider Finder (6%, 9 Votes)

Total Voters: **147**

LIKE US ON FACEBOOK

RECENT POSTS

- Camera Translator All Language App
- Why MVVM and How to execute MVVM combined with Data Binding and LiveData in Four Simple Steps
- Camera Translator All Languages – Android Project idea source code for students
- Note Scan Android App -Project idea source code for students
- How To Build Android WhatsApp Clone Chat Application
- Android Car Rental App with Web Admin Panel – Project Idea Source Code For Students

Target SDK 23

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: AndroidLocation

Package: com.inducesmile.androidlocation

Select Blank Activity

Keep other default selections

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

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.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <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 Location API</string
```

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

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.androidlocation">
    <uses-permission android:name="android.permission.AC
    <uses-permission android:name="android.permission.AC
    <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
                <category android:name="android.intent.c
            </intent-filter>
        </activity>
    </application>
</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 .

```
    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:textStyle="bold"
                android:textColor="@color/colorBlack"
                android:layout_marginTop="32dp"
                android:layout_gravity="center"/>
            <TextView
                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: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:text="@string/city"
            android:textSize="18dp"
            android:textStyle="bold"
            android:textColor="@color/colorBlack"
            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>
```

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

We are going to use the following line of code to obtain the LocationManager instance.

locationManager = (LocationManager) getSystemService(Service.LOCATION_SERVICE);

If you are using android API Level 23, the app will always request for user permission whenever the user current or last known location is requested.

The code below is use for the user request.

```
if (ActivityCompat.checkSelfPermission(this, Manifest.pe
    ActivityCompat.requestPermissions(MainActivity.this,
} else {
    locationManager.requestLocationUpdates(LocationManag
    if (locationManager != null) {
        location = locationManager.getLastKnownLocation(
    }
}
```

For us to use the obtained longitude and latitude values to obtain the address location, we will create an inner Handler class.

This Handler instance will communicate between the Thread that runs the GeoCoder class and the Views in the UI Thread.

Copy and paste the complete code for the MainActivity.java file.

```
import android.Manifest;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.widget.TextView;
import java.io.IOException;
import java.util.List;
import java.util.Locale;
public class MainActivity extends AppCompatActivity impl
    private TextView latitudePosition;
    private TextView longitudePosition;
    private TextView currentCity;
    private LocationManager locationManager;
    private Location location;
    private final int REQUEST_LOCATION = 200;
    private static final String TAG = "MainActivity";
    @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)
        locationManager = (LocationManager) getSystemSer
        if (ActivityCompat.checkSelfPermission(this, Man
            ActivityCompat.requestPermissions(MainActivi
        } else {
            locationManager.requestLocationUpdates(Locat
            if (locationManager != null) {
                location = locationManager.getLastKnownL
            }
        }
        if (locationManager.isProviderEnabled(LocationMa
            if (location != null) {
                latitudePosition.setText(String.valueOf(
                longitudePosition.setText(String.valueOf
                getAddressFromLocation(location, getAppl
            }
        } else {
```

```
    public void onLocationChanged(Location location) {
        latitudePosition.setText(String.valueOf(location
        longitudePosition.setText(String.valueOf(locatio
        getAddressFromLocation(location, getApplicationC
    }
    @Override
    public void onStatusChanged(String provider, int sta
    }
    @Override
    public void onProviderEnabled(String provider) {
    }
    @Override
    public void onProviderDisabled(String provider) {
        if (provider.equals(LocationManager.GPS_PROVIDER
            showGPSDisabledAlertToUser();
        }
    }
    @Override
    public void onRequestPermissionsResult(int requestCo
        if (requestCode == REQUEST_LOCATION) {
            if (grantResults.length == 1 && grantResults
            }
        }
    }
    private void showGPSDisabledAlertToUser() {
        AlertDialog.Builder alertDialogBuilder = new Ale
        alertDialogBuilder.setMessage("GPS is disabled i
                .setCancelable(false)
                .setPositiveButton("Goto Settings Page T
                    public void onClick(DialogInterface
                        Intent callGPSSettingIntent = ne
                        startActivity(callGPSSettingInte
                    }
                });
        alertDialogBuilder.setNegativeButton("Cancel", n
            public void onClick(DialogInterface dialog,
                dialog.cancel();
            }
        });
        AlertDialog alert = alertDialogBuilder.create();
        alert.show();
    }
```

```
                String result = null;
                try {
                    List<Address> list = geocoder.getFro
                            location.getLatitude(), loca
                    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);
                    } else
                        msg.what = 0;
                    msg.sendToTarget();
                }
            }
        };
        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);
        }
    }
```

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 among the first to receive my new post once it is published.

Please if you love this tutorial, kindly download my new android app – ***Daily Calculator Tool*** – in Google Play Store and let me know what you think about it.

OTHER INTERESTING POSTS:

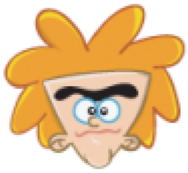
ANDROID TUTORIAL PART 4: AFRICAN HAIRSTYLE APP

Android ToolBar With Tab and Horizontal Swipe View

A simple Android Number Matching Game Tutorial

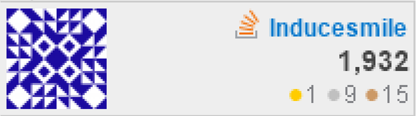
Android FirebaseUI Authentication Example

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



Inducesmile

1,932

● 1 ● 9 ● 15

ADD A COMMENT

You must be logged in to post a comment.

Accept