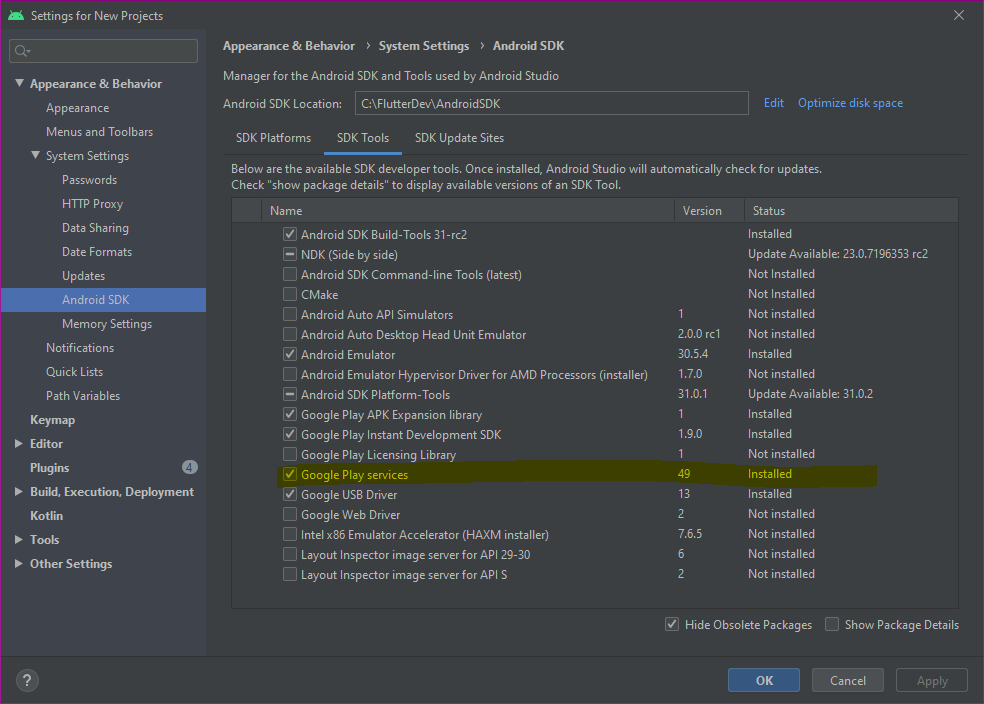
# Maps Current Location App

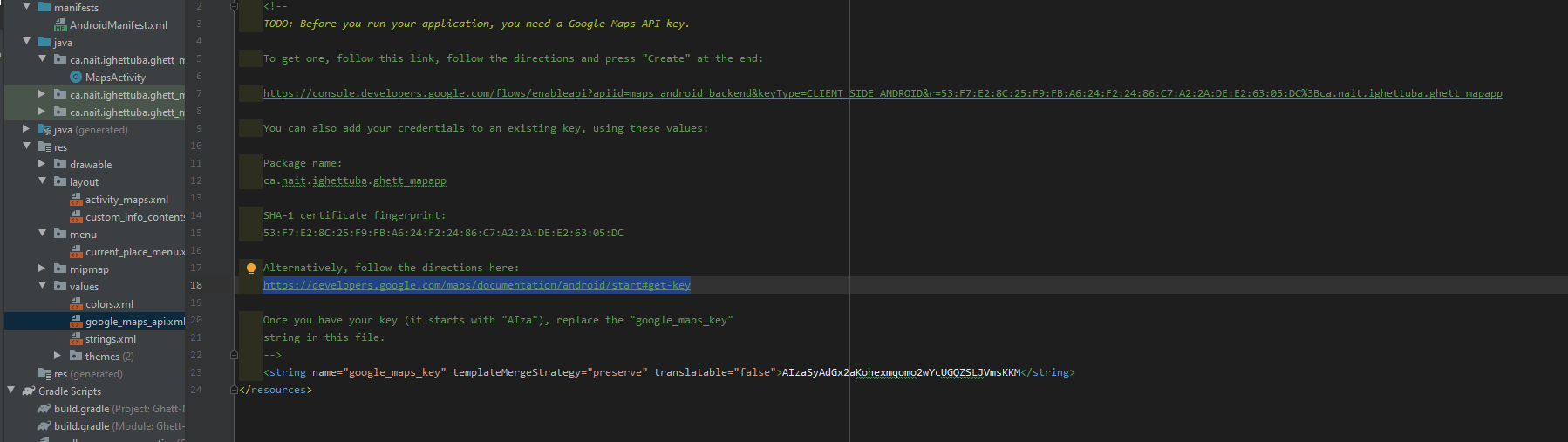
1.Update SDK Manager to include Google Play Services Package.



2. Create Google Maps Activity and name it

3. In res > values > google\_maps\_api.xml

1. Before you run your application, you need a Google Maps API key. Visit this site and follow instructions to get the key.
   1. <https://developers.google.com/maps/documentation/android/start#get-key>
   2. Add API key to app in google\_maps\_api.xml



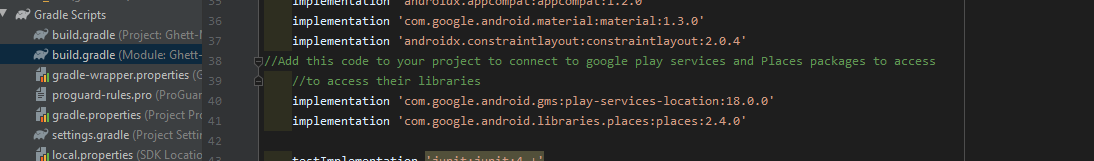
4. Instantiate Places API Client

a. Add this to .graddle file (app level)

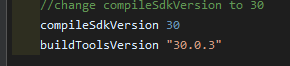
i. Allows access to Google Play Libraries that will be needed

1) implementation'com.google.android.gms:play-services-location:18.0.0'

2) implementation'com.google.android.libraries.places:places:2.4.0'



ii. Check your compileSdkVersion clause. Set to 30

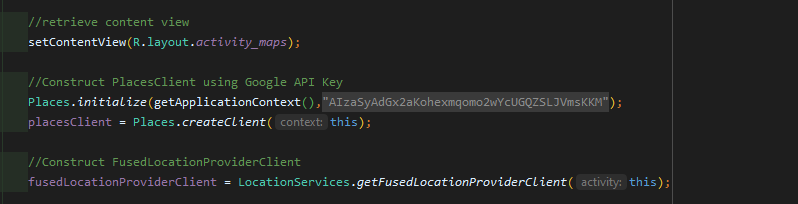


5. OnCreate Method()

a. Get layout view

b. Initialize placesCLient using API key(stored in google\_maps\_api.xml)

c. Construct FusedLocationProviderClient



6. Request Location permission

a. in order to determine the location of the device and to allow the user to tap the My Location button on the map.

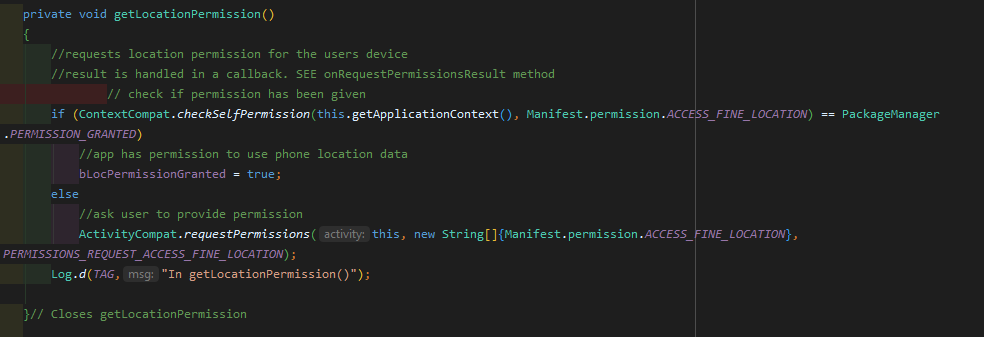
b. Done in manifest.xml



7. getLocationPermission() method

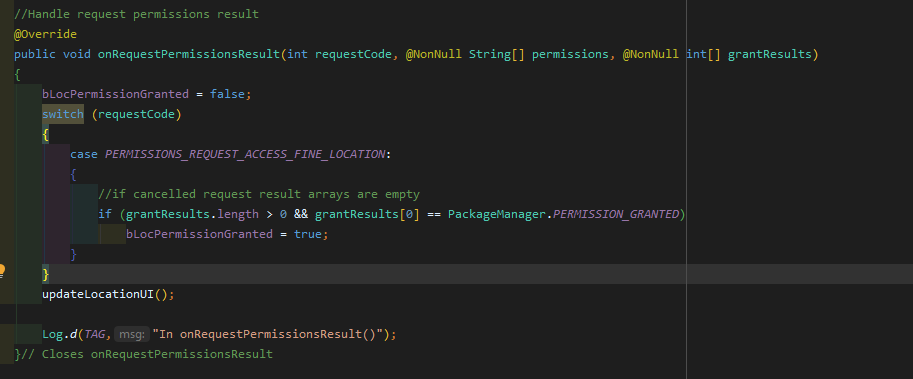
a. User can allow or deny location permission.

b. This code checks if permission is given.. If not, it is requested.



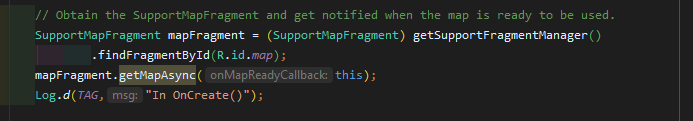
8. Overide onRequestPermissionsResult()

a. To handle result of permission request



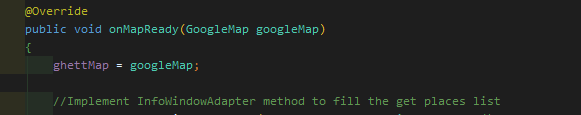
9. Display a map using the Maps SDK for android

a. Add fragment element to activity\_maps.xml (might be added on create activity)



11. Implement onMapReadyCallback()

a. Set map up when the google object is available



12. In onCreate(), get fragment view

a. Use getMapAsync() to register for the map callback



13. UpdateLocationUI()

a. Set controls on the map

b. If permission granted

i. Enable location layer and control

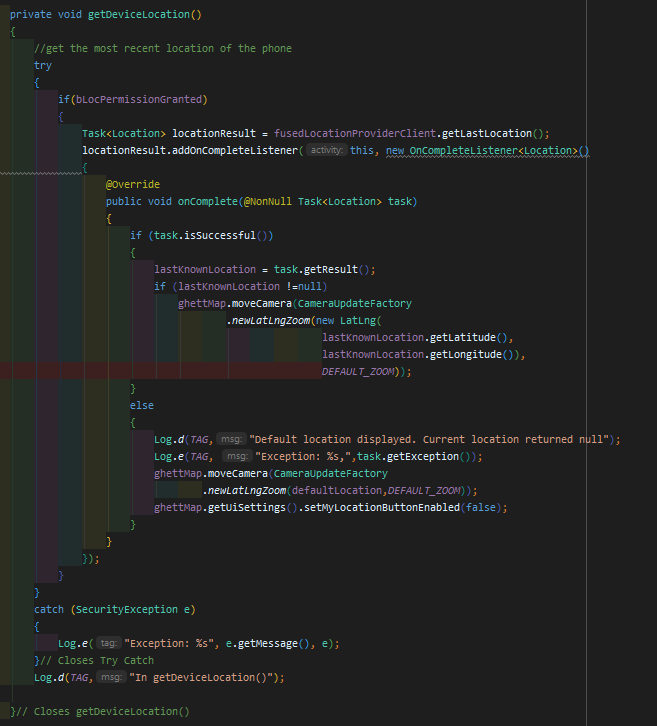
ii. Otherwise disable layer and control, and set current location to null



14. Get DeviceLocation()

a. Get the device location and position on the map

b. Using the fused location provider in google play services



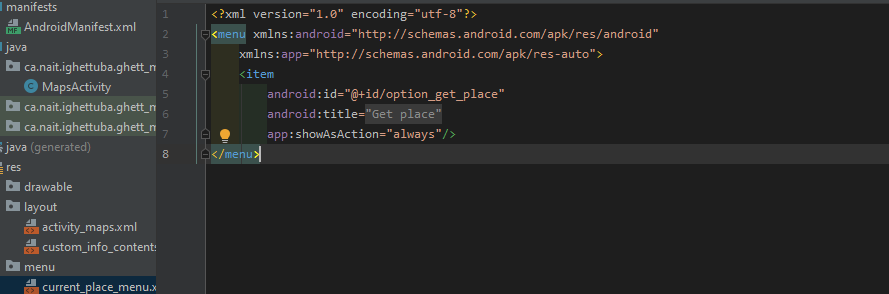
15. Get the current place

a. Using Places SDK for list of likely places ie

i. Businesses

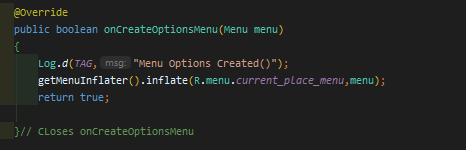
ii. Point of interest

b. Create current\_place\_xml



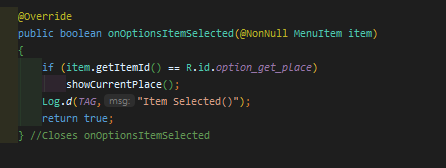
i. Override onCreateOptionsMenu()

1) Inflate menu with current\_place\_menu.xml



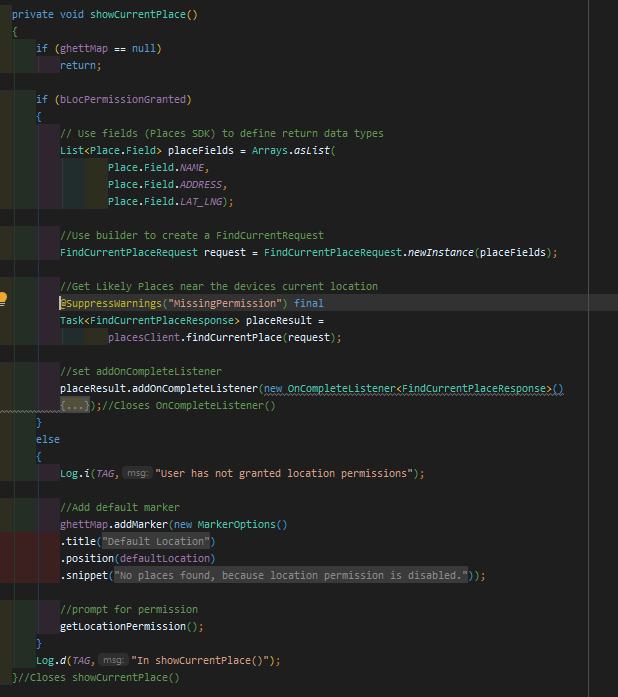
ii. Override onOptionsItemSelected()

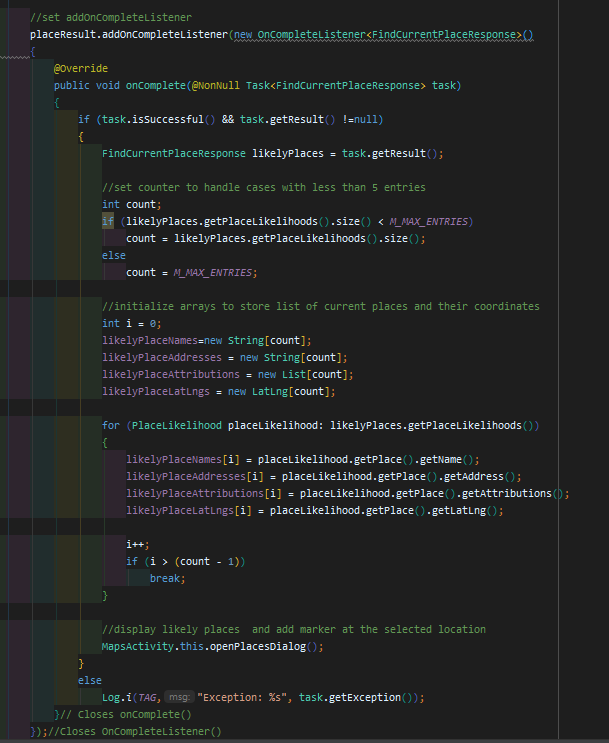
1) Get the current place when user selects the Get Place Option



c. showCurrentPlace() methhod

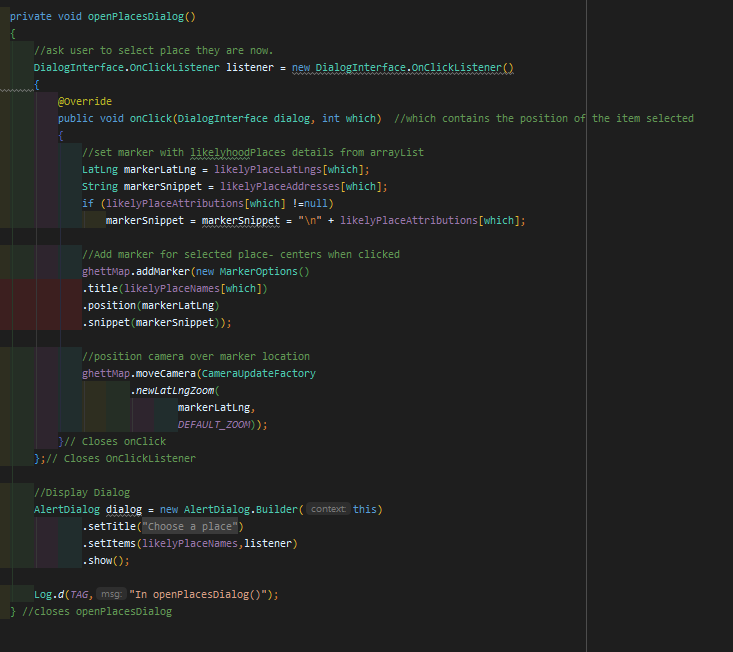
i. Return a list of likely places around the devices location





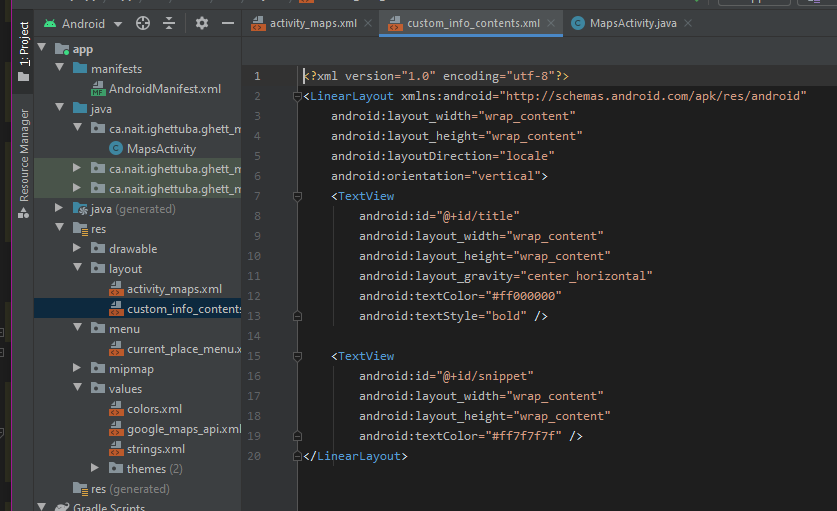
ii. openPlacesDialog() method

1) Display a list of places for the user to select from



iii. Create a custom layout for displaying the places information.

1) Custom\_info\_xml



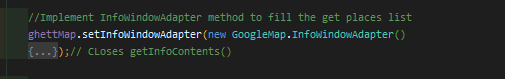
2) 2 textveiws

a) Title

b) Snippet/info

d. Implement InfoWindowAdapter() in onMapReady() method

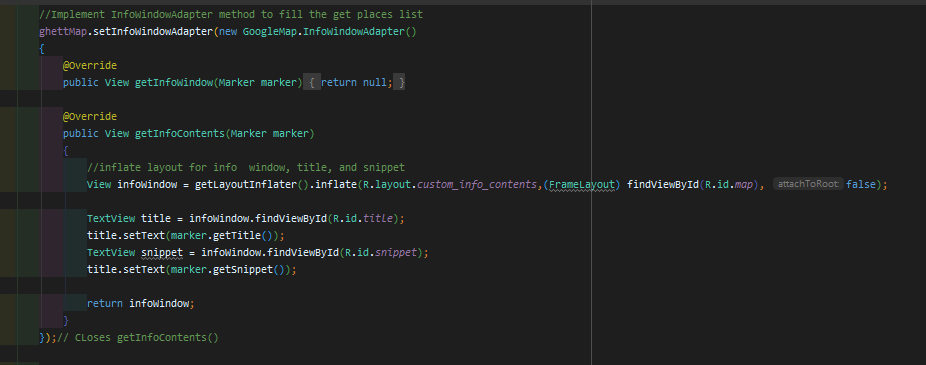
i. Inflate custom\_info\_contents layout with info from liklihood places



ii. 2 Override methods

1) getInfoWindow()

2) getInfoContents()



e. Save the map's state

i. Save camera position and device location.

1) When user rotates phone or changes phone config. Android framework destroys and rebuilds map activity.

a) Smooth User Experience

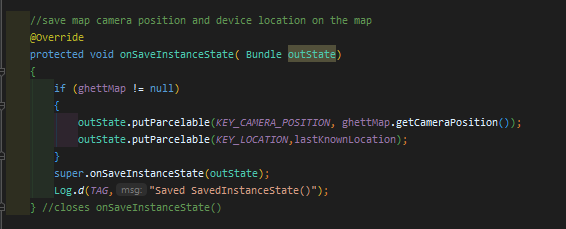
b) Good practices

ii. Create variables to store key values.

1) KEY\_CAMERA\_POSITION , KEY\_LOCATION

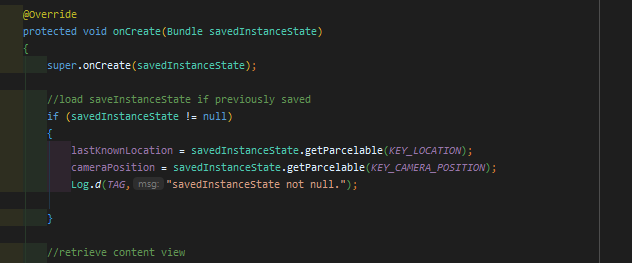
iii. Implement onSaveInstanceState() method after onCreate()

1) Save when activity pauses



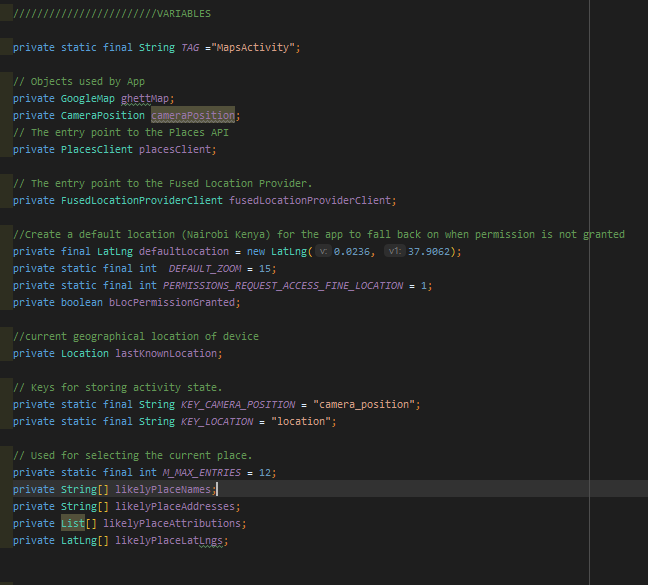
iv. In onCreate() method

1. Retrieve device location and camera position



CONGRATS!!!!!!! YOU HAVE A WORKING MAPS APP!!!!!!

### VARIABLES USED



# Resources

The following link will take you to an android page that I followed to create my App. Al coding was done by hand (no copy/paste situation). The tutorial is older than the version of Android studio I used.

* https://developers.google.com/maps/documentation/android-sdk/current-place-tutorial