

## Family Map Client - Google Map Fragment Tutorial

*The following tutorial assumes you have an existing Android Studio project and have passed off the Login assignment.*

### Setup Your Project to Use Google Maps

1. **Add Google Play Services to Android Studio and to your project** using the following guide from our friends at Google: <https://developers.google.com/android/guides/setup>

When you are finished with the above you should have completed the following:

- a. Added the Google Repository to the Android Studio SDK Manager
  - b. Added a dependency on the latest version of Google Play Services to your app-level build.gradle file. Make sure your Play Services version syncs properly with gradle. If it doesn't, you may need to downgrade the version manually.
2. **Create a Google Maps Activity.**
    - a. Right click on the package in your project where you have your Activities. Navigate to New->Google->Google Maps Activity.
    - b. Click Finish. This will generate MapsActivity.java and activity\_maps.xml that we will delete later.
  3. **Obtain your API key.** The previous step created a file called google\_maps\_api.xml. This is a resource file that will contain your API key. Now, you must **obtain your API key**. An API key is a \*cough\* key that allows you to access Google's maps API. Follow these steps to get one:
    - a. Copy/paste into your browser the web link provided for you in google\_maps\_api.xml. It is likely is on line 7. This will take you to the Google Developers Console.
    - b. Select "Create a project" and click Continue. Wait for the next page to load.
    - c. Click "Create API key"
    - d. Copy the API key (a long string of seemingly random characters).
    - e. Click "Restrict Key" and restrict it to Android apps. Click save.
    - f. Paste your key into your google\_maps\_api.xml, replacing the text that says YOUR\_KEY\_HERE

### Setup Your MapFragment

1. Modify your activity\_map.xml created in step 2b above:
  - a. Rename the file to fragment\_map.xml for consistency
  - b. Change the contents of the file to the following. This gives you a LinearLayout with a MapView widget inside:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <com.google.android.gms.maps.MapView
        android:id="@+id/mapview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>

```

2. Delete MapActivity.java.
3. Create the new MapFragment
  - a. Right click on the package in your project where you have your Activities. Navigate to New->Activity->Empty Activity
  - b. Set the activity name to "MapFragment"
  - c. Uncheck the "Generate Layout File" box, as you will use fragment\_map.xml that you already have
  - d. Click Finish
4. Modify MapFragment. You should only have the onCreate method so far.
  - a. Change the extension from "extends AppCompatActivity" to "extends SupportMapFragment"
  - b. Make the class implement OnMapReadyCallback. Override the onMapReady method. Tip: This method gets called after the map is loaded, and is where you will draw your markers and lines at a later time.
  - c. Change onCreate from protected to public. Implement and Override the onCreateView method
  - d. Add two private fields to the class: GoogleMap and MapView
5. Modify onCreateView. Follow these steps *exactly*:
  - a. On the first line of onCreateView, call super.onCreateView with the appropriate parameters.
  - b. Next, View v = inflater.inflate(R.layout.fragment\_map, ViewGroup, false);
  - c. Next, instantiate your MapView field from step 4d referencing the MapView widget in your fragment\_map.xml
  - d. Using your MapView object, call mapView.onCreate, mapView.onResume, and mapView.getMapAsync(this) *in that order*.
  - e. Change the return value to "v" from step 5b.
6. Modify your MainActivity to start your MapFragment at the appropriate time using your FragmentManager
7. Run your app.