

1. Add textbox

2. Send textbox data through intent

```
EditText editText = (EditText) findViewById(R.id.editText1);  
String message = editText.getText().toString();  
intent.putExtra(EXTRA_MESSAGE, message);
```

3. Get intent data and display

```
Intent intent = getIntent();  
String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);  
TextView textView = new TextView(this);  
textView.setTextSize(40);  
textView.setText(message);  
  
// Set the text view as the activity layout  
setContentView(textView);
```

4. Get google play services

<http://developer.android.com/google/play-services/setup.html>

5. Add project and get API key

<https://code.google.com/apis/console/?noredirect&pli=1#project:980860515645>

6. Add to manifest

```
<!-- Google Maps -->  
<!-- This is the Google Maps API key. Add this to your own manifest  
to take advantage of google maps -->  
    <meta-data android:name="com.google.android.maps.v2.API_KEY"  
        android:value="AlzaSyDpQX2FEHJQujTgpcN60tIPYlqXCerm4Ac"/>  
    <meta-data  
        android:name="com.google.android.gms.version"  
        android:value="@integer/google_play_services_version" />
```

7. Add permission requests

```
<!-- Permission requests -->  
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>  
    <uses-permission android:name="android.permission.INTERNET"/>  
    <uses-permission  
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
```

<!-- My Location -->

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"/>

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>

9. Create new gmap activity

10. Edit xml

```

<?xml version="1.0" encoding="utf-8"?><LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <!-- This fragment is where the Google map will be located -->
    <fragment

        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        class="com.google.android.gms.maps.SupportMapFragment"/>

</LinearLayout>

```

## 11. Edit .java

```

-----
package com.example.coolapp;

/*These imports are all necessary
 * You also must add the Google-play-services_lib library to your project
 * Check out http://developer.android.com/google/play-services/setup.html
 * for instructions
 *
 * Outside of Eclipse you must get a Google API key for Google maps and place
 * it in your project manifest. Check out this link:
 * https://developers.google.com/maps/documentation/android/start
 * and this project manifest for examples
 *
 * Other than this .java page, you must have a .xml layout page. Find it under
 * layout/activity_google_maps.xml
 */
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.SupportMapFragment;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;

public class GoogleMapsSample extends FragmentActivity

```

```

{

private GoogleMap sampleMap; //Google Map variable


@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    //set xml layout
    setContentView(R.layout.activity_google_maps);
}

@Override
protected void onResume() {
    super.onResume();
    setUpMapIfNeeded();
}


private void setUpMapIfNeeded() {
    // Do a null check to confirm that we have not already instantiated the map.
    if (sampleMap == null) {
        // Try to obtain the map from the SupportMapFragment.
        sampleMap = ((SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map))
        .getMap();
        // Check if we were successful in obtaining the map.
        if (sampleMap != null) {
            //Allow user to pinpoint their location
            sampleMap.setMyLocationEnabled(true);
        }
    }
}

}

```