

Google Maps is one of the many applications bundled with the Android platform. In addition to

simply using the Maps application, you can also embed it into your own applications and make it do

some very cool things. This section describes how to use Google Maps in your Android applications

and programmatically perform the following:

- Change the views of Google Maps.
- Obtain the latitude and longitude of locations in Google Maps.
- Perform geocoding and reverse geocoding (translating an address to latitude and longitude and vice versa).

Creating the Project

To get started, you need to first create an Android project so that you can display Google Maps in

your activity:

1. Using Android Studio, create an Android project and name it LBS.
2. From the Create New Project Wizard, select Google Maps Activity

Obtaining the Maps API Key

Beginning with the Android SDK release v1.0, you need to apply for a free Google Maps API key before you can integrate Google Maps into your Android application. When you apply for the key, you must also agree to Google's terms of use, so be sure to read them carefully.

To get a Google Maps key, open the **google_maps_api.xml** file that was created in your LBS project. Within this file is a link to create a new Google Maps key. Simply copy and paste the link into your browser and follow the instructions. Make note of the key that Google gives you because you need it later in this project.

MapsActivity.java file:

```
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
```

```
import com.google.android.gms.maps.model.MarkerOptions;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified
        // when the map is ready to be used.
        SupportMapFragment mapFragment =
            (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    @Override

    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        LatLng boston = new LatLng(42.3601, -71.0589);
        mMap.addMarker(new MarkerOptions().position(boston).title(
            "Boston, Mass"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(boston));
    }
}
```