Intro to Maps V2

Maps: Not Invented in 2008!

- Android 1.0: Maps SDK Add-On
  - Same as what Google Maps app used
  - Creaky API, particularly with fragments
- The Fork
  - Google Maps started using own engine
  - Gained more features than old SDK add-on supported

Maps: Not Invented In 2012 Either!

- Maps V2
  - Introduced in late 2012
  - New API, new device requirements, new features
    - Close to full Google Maps app
- Maps V1: Deprecated!
  - Can no longer official get Maps V1 API keys
  - Still runs, if you have keys from before

Legal, Without Ease

- Maps V1 and Maps V2: Terms of Service
  - Required to use these APIs
  - Put restrictions on how you use those APIs
    - Example: no real-time turn-by-turn navigation
  - IANALNDIPOOTV
    - Please have qualified legal counsel review these terms if they concern you

Users Have Issues (No, Really!)

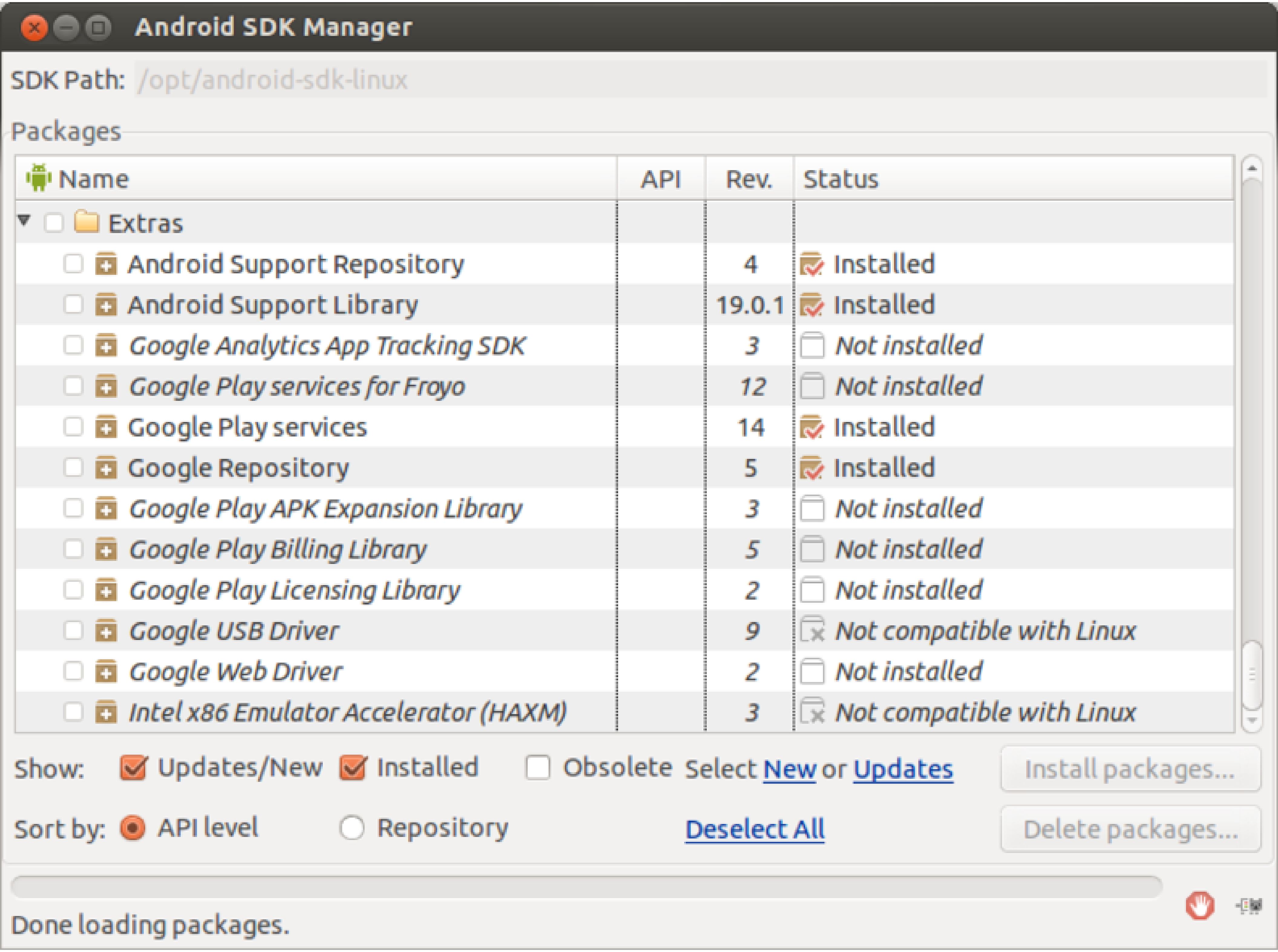
- Requirements
  - Play Services Framework
    - May already exist
    - May be available from the Play Store
    - Not available by other legitimate means
  - OpenGL ES 2.0
    - On 99.9% of devices with the Play Store

You Have Issues Too

- Limited Emulator Support
  - Play Services Framework not available in most emulator images
  - Play Services Framework cannot be updated
- Expectation: Test on Hardware

Preparation M

- Setting Up Maps V2 Development
  - Download Google Play Services SDK from SDK Manager (in Extras area)
  - Result: Android Library Project
    - Eclipse: import into workspace, then attach to application project
    - Gradle: `compile 'com.google.android.gms:play-services:4.1.32'`
    - Legacy command-line: **`android update project`**
    - Other IDEs: ???



Android SDK Manager

SDK Path: /opt/android-sdk-linux

Packages

Name	API	Rev.	Status
▾ □ Extras			
□ Android Support Repository		4	Installed
□ Android Support Library		19.0.1	Installed
□ *Google Analytics App Tracking SDK*		*3*	*Not installed*
□ *Google Play services for Froyo*		*12*	*Not installed*
□ Google Play services		14	Installed
□ Google Repository		5	Installed
□ *Google Play APK Expansion Library*		*3*	*Not installed*
□ *Google Play Billing Library*		*5*	*Not installed*
□ *Google Play Licensing Library*		*2*	*Not installed*
□ *Google USB Driver*		*9*	*Not compatible with Linux*
□ *Google Web Driver*		*2*	*Not installed*
□ *Intel x86 Emulator Accelerator (HAXM)*		*3*	*Not compatible with Linux*

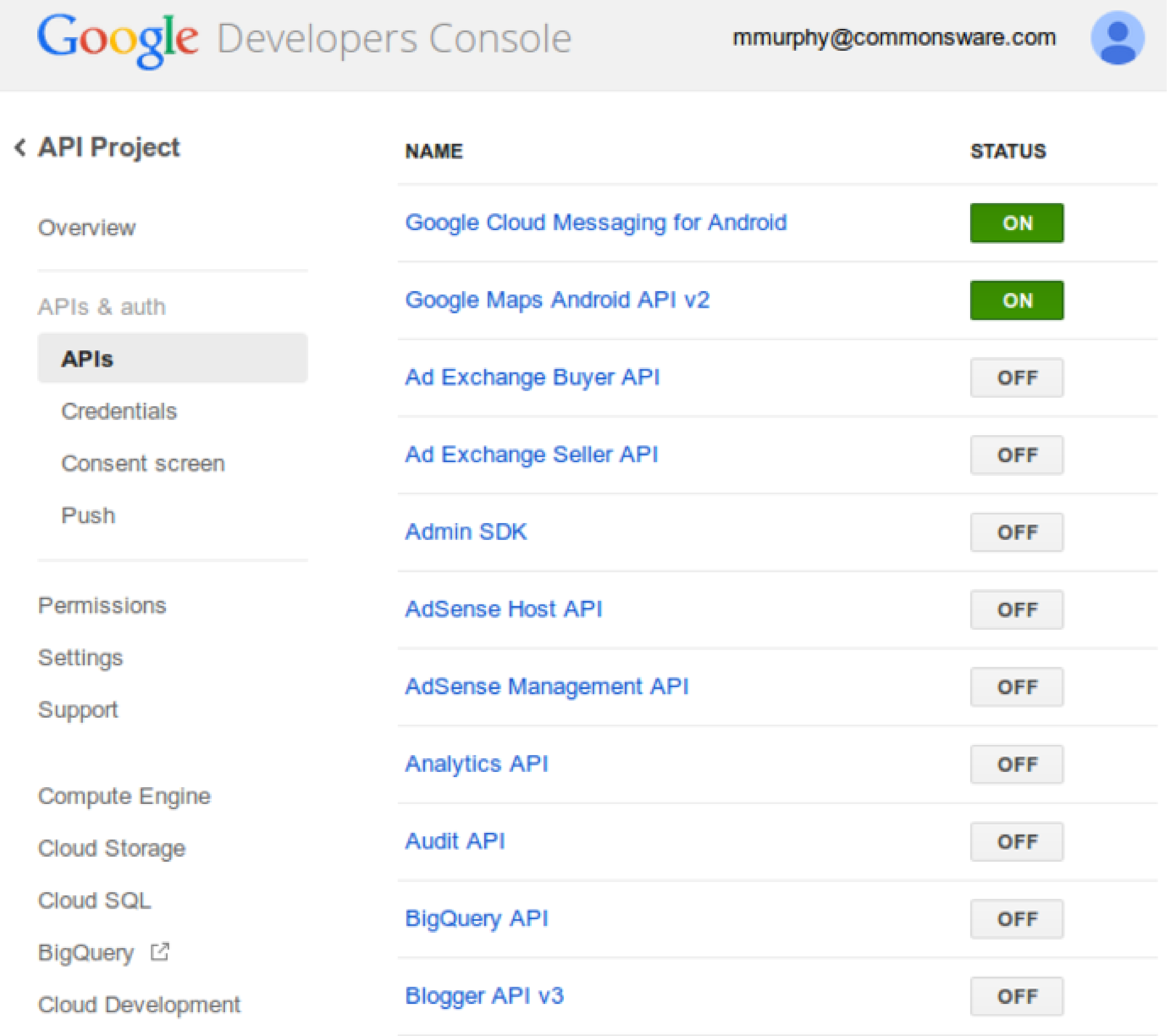
Show: ☑ Updates/New ☑ Installed □ Obsolete Select New or Updates

Sort by: ● API level ○ Repository Deselect All

Done loading packages.

Yet More Preparation M

- Maps V2 API Key
  - No, it's not the same as your Maps V1 API key
  - Obtain it from the Google API Console
  - Tied to your package name, signing key
    - SHA-1 fingerprint vs. MD5
    - Up to ~5 signing key/package name pairs per API key



< API Project

Overview

APIs & auth

APIs

Credentials

Consent screen

Push

Permissions

Settings

Support

Compute Engine

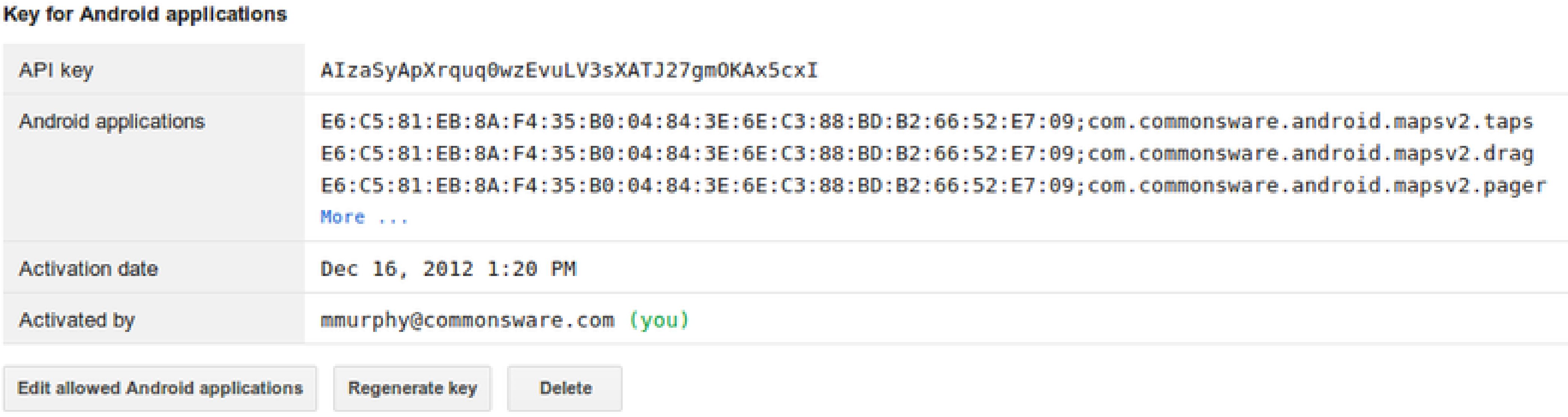
Cloud Storage

Cloud SQL

BigQuery

Cloud Development

NAME	STATUS
Google Cloud Messaging for Android	ON
Google Maps Android API v2	ON
Ad Exchange Buyer API	OFF
Ad Exchange Seller API	OFF
Admin SDK	OFF
AdSense Host API	OFF
AdSense Management API	OFF
Analytics API	OFF
Audit API	OFF
BigQuery API	OFF
Blogger API v3	OFF

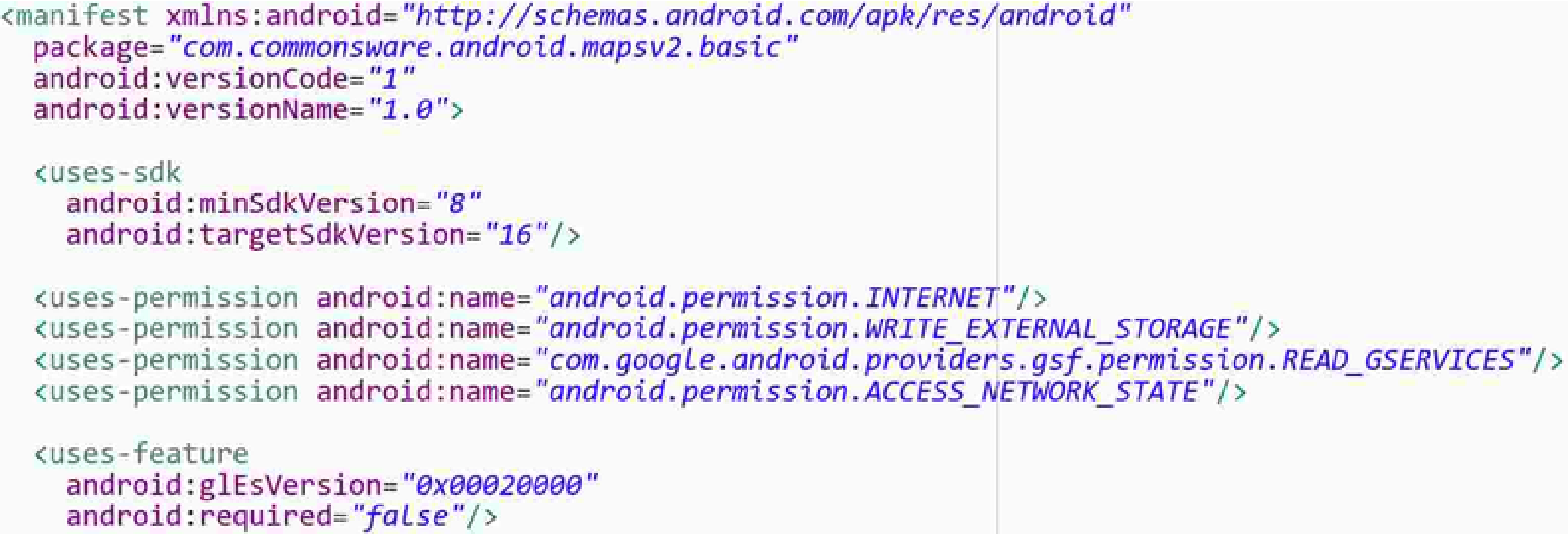


Key for Android applications

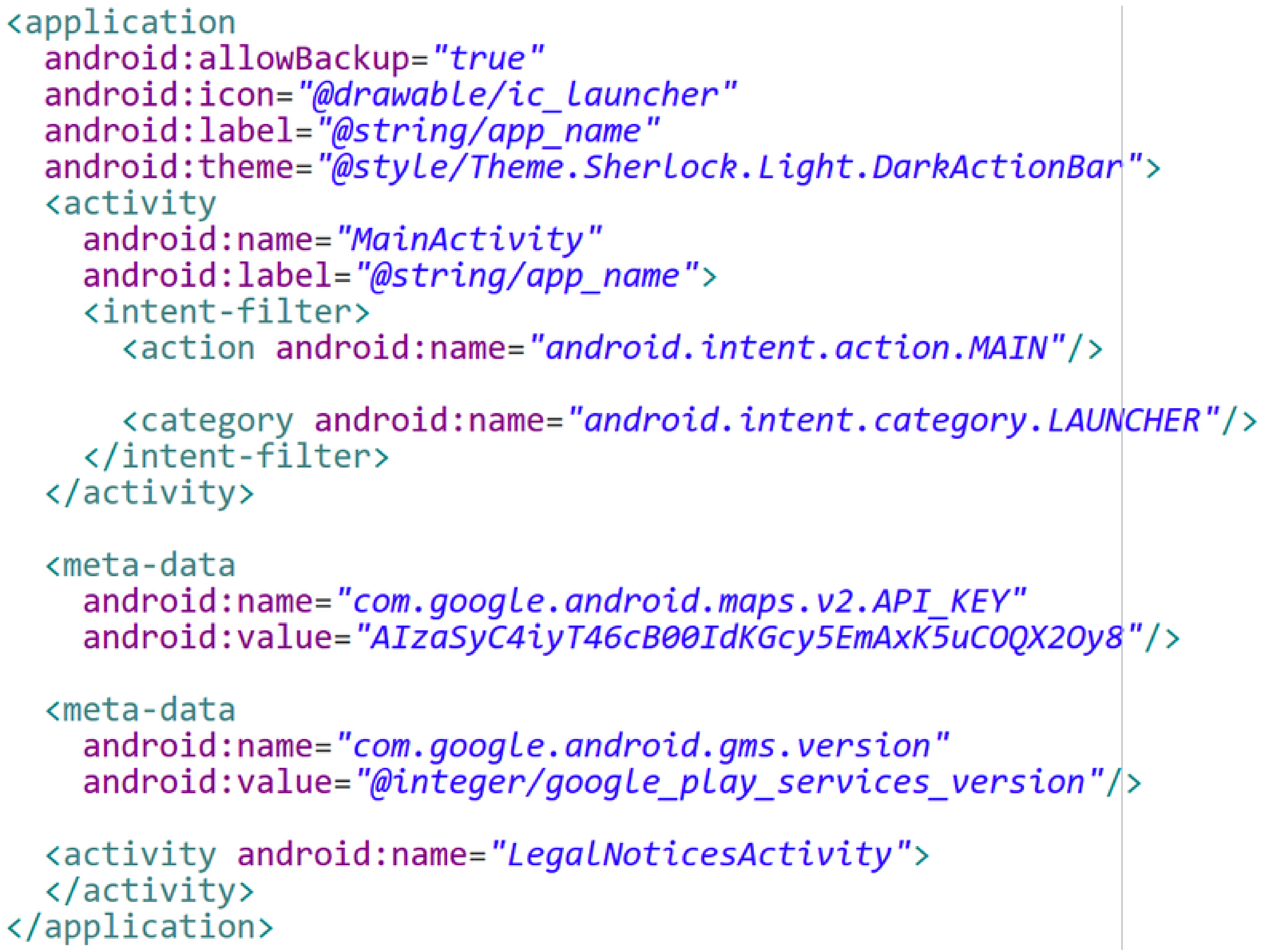
API key	AIzaSyApXrquq0wzEvuLV3sXATJ27gm0KAx5cxI
Android applications	E6:C5:81:EB:8A:F4:35:B0:04:84:3E:6E:C3:88:BD:B2:66:52:E7:09;com.commonsware.android.mapsv2.taps E6:C5:81:EB:8A:F4:35:B0:04:84:3E:6E:C3:88:BD:B2:66:52:E7:09;com.commonsware.android.mapsv2.drag E6:C5:81:EB:8A:F4:35:B0:04:84:3E:6E:C3:88:BD:B2:66:52:E7:09;com.commonsware.android.mapsv2.pager More ...
Activation date	Dec 16, 2012 1:20 PM
Activated by	mmurphy@commonsware.com (you)

Let's Get Mappy, People

- A Maps V2 Project
  - ...does not need to use a Google APIs SDK as a build target
  - ...does need to restrict to OpenGL ES 2.0 devices
    - Or make optional, but test at runtime
  - ...does need a few permissions
  - ...does need a Maps V2 API key
  - ...does need to specify a Google Play Services version
    - Identifies version of the SDK you build against



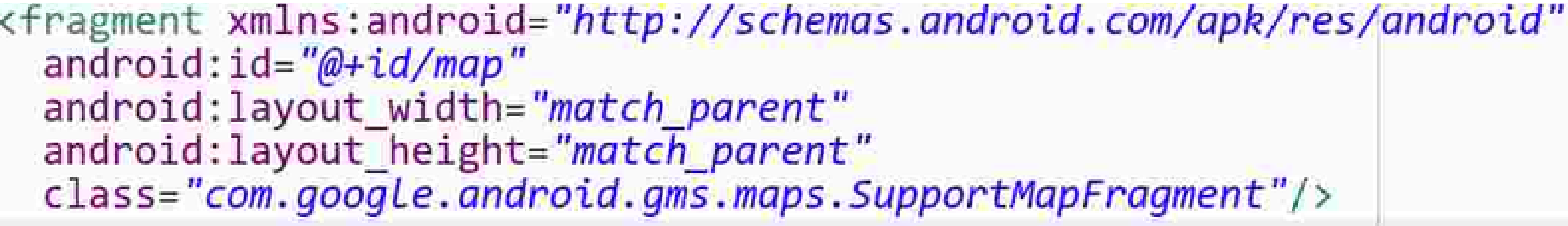
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.commonsware.android.mapsv2.basic"
  android:versionCode="1"
  android:versionName="1.0">

  <uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="16"/>

  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
  <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>

  <uses-feature
    android:glEsVersion="0x00020000"
    android:required="false"/>
```



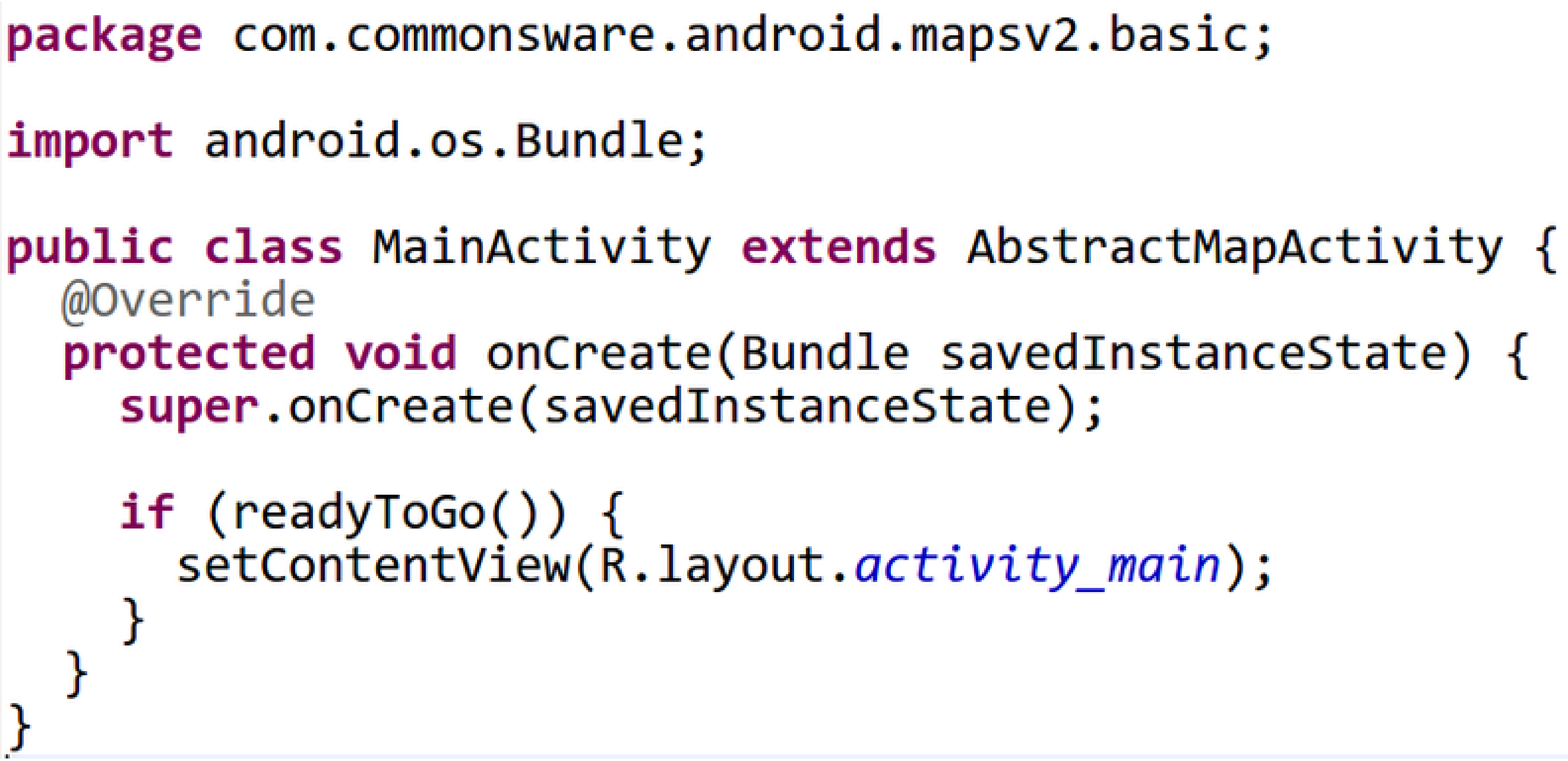
```
<application
  android:allowBackup="true"
  android:icon="@drawable/ic_launcher"
  android:label="@string/app_name"
  android:theme="@style/Theme.Sherlock.Light.DarkActionBar">
  <activity
    android:name="MainActivity"
    android:label="@string/app_name">
    <intent-filter>
      <action android:name="android.intent.action.MAIN"/>

      <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
  </activity>

  <meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyC4iyT46cB00IdKGcy5EmAxK5uCOQX2Oy8"/>

  <meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version"/>

  <activity android:name="LegalNoticesActivity">
  </activity>
</application>
```

Let's Get Mappy, People

- A Maps V2 Activity
  - …does not have to inherit from any magic activity class
  - …does need to use MapFragment, SupportMapFragment, or MapView
  - …does have a lot of legwork to ensure Maps V2 is available, the user has access to license terms, etc.
    - …all of which can be nicely abstracted away



```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/map"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  class="com.google.android.gms.maps.SupportMapFragment"/>
```



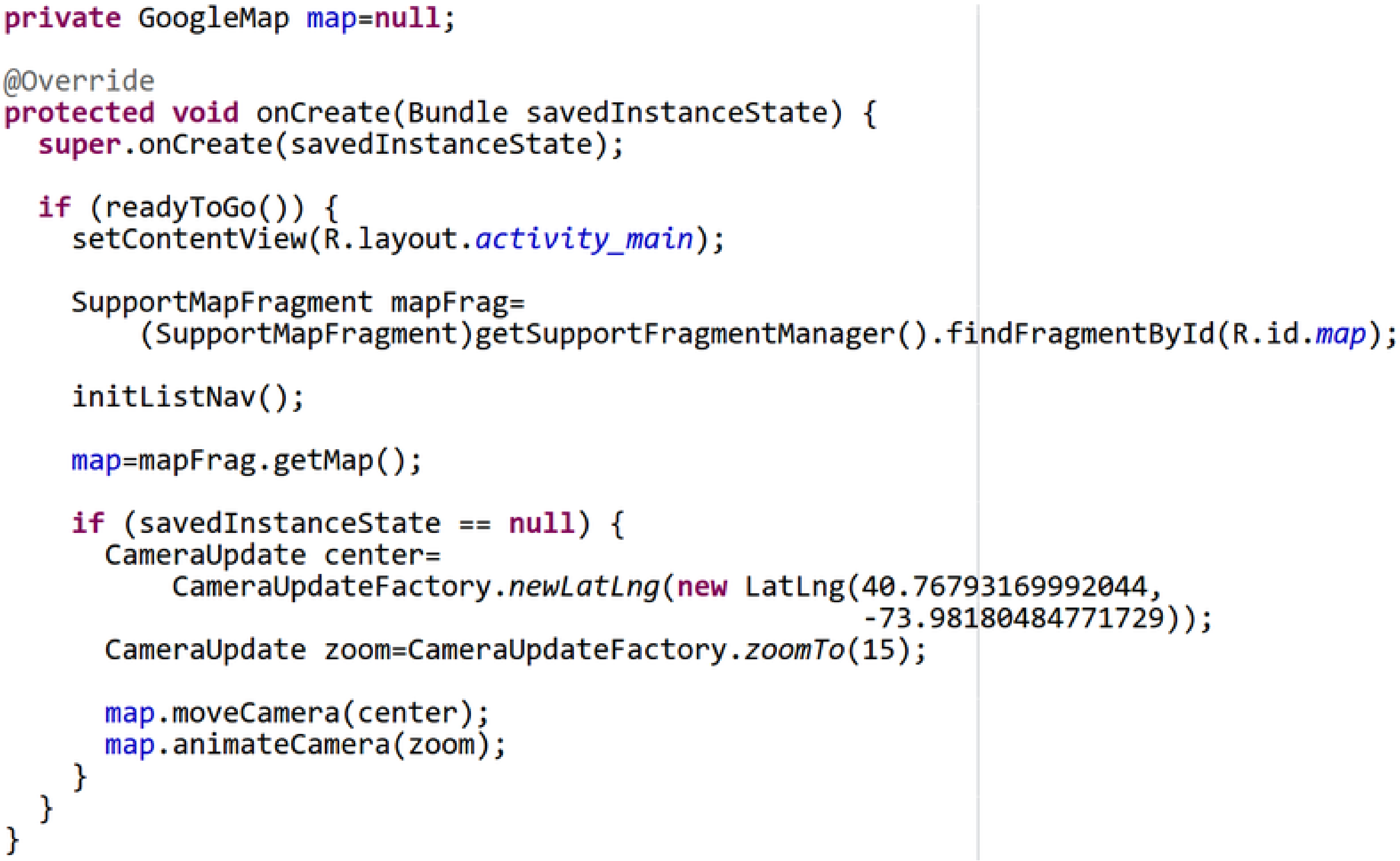
```
package com.commonsware.android.mapsv2.basic;

import android.os.Bundle;

public class MainActivity extends AbstractMapActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    if (readyToGo()) {
      setContentView(R.layout.activity_main);
    }
  }
}
```

Working (with a map) For a Living

- GoogleMap
  - Your gateway to the actual mapping API
  - Obtained from MapFragment, SupportMapFragment, or MapView via `getMap()`
  - Most method calls result in IPC
    - Comparatively slow
    - Avoid doing lots of these in a tight loop
    - Must be called on the main application thread

Showing Something Useful

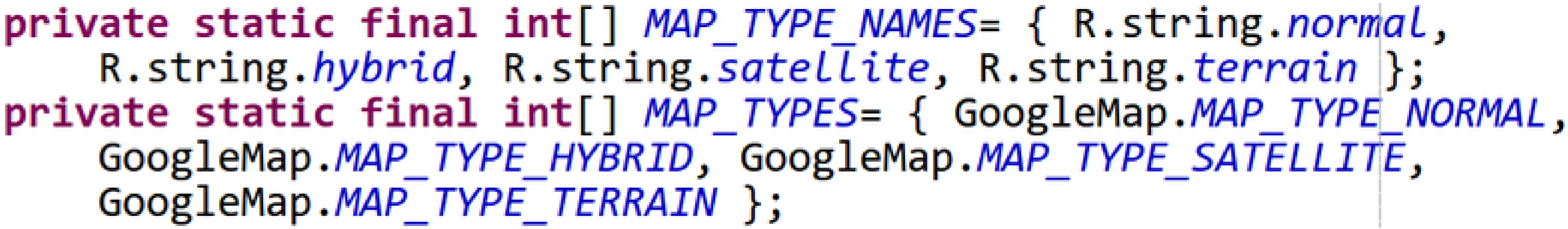
- CameraUpdate
  - Changes in characteristics of the “camera”
    - Center, zoom, etc.
  - Create using CameraUpdateFactory
  - Apply to GoogleMap
    - `moveCamera()`
    - `animateCamera()`



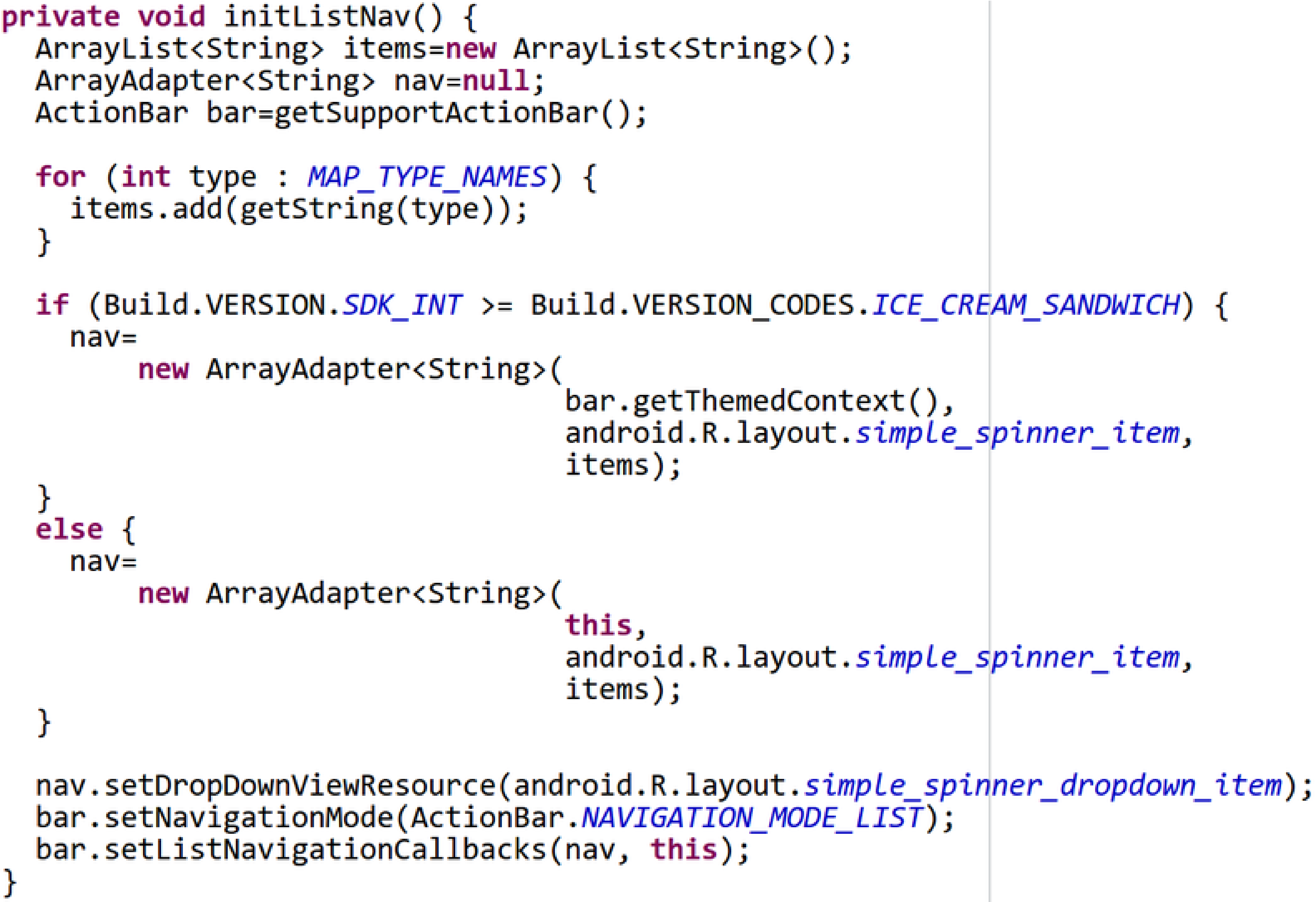
```
private GoogleMap map=null;

@Override
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);

  if (readyToGo()) {
    setContentView(R.layout.activity_main);

    SupportMapFragment mapFrag=
        (SupportMapFragment)getSupportFragmentManager().findFragmentById(R.id.map);

    initListNav();

    map=mapFrag.getMap();

    if (savedInstanceState == null) {
      CameraUpdate center=
          CameraUpdateFactory.newLatLng(new LatLng(40.76793169992044,
                                                   -73.98180484771729));
      CameraUpdate zoom=CameraUpdateFactory.zoomTo(15);

      map.moveCamera(center);
      map.animateCamera(zoom);
    }
  }
}
```

Typing Up a Map

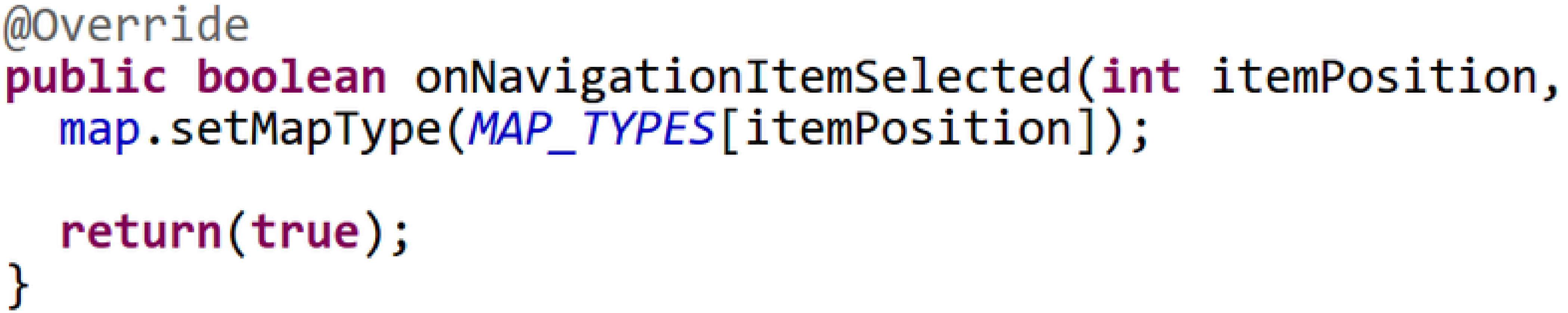
- Default Map Type: `MAP_TYPE_NORMAL`
- Changing Map Types
  - Choose a type (e.g., `MAP_TYPE_HYBRID`)
  - Call `setMapType()` on your GoogleMap



```java
private static final int[] MAP_TYPE_NAMES= { R.string.normal,
    R.string.hybrid, R.string.satellite, R.string.terrain };
private static final int[] MAP_TYPES= { GoogleMap.MAP_TYPE_NORMAL,
    GoogleMap.MAP_TYPE_HYBRID, GoogleMap.MAP_TYPE_SATELLITE,
    GoogleMap.MAP_TYPE_TERRAIN };
```



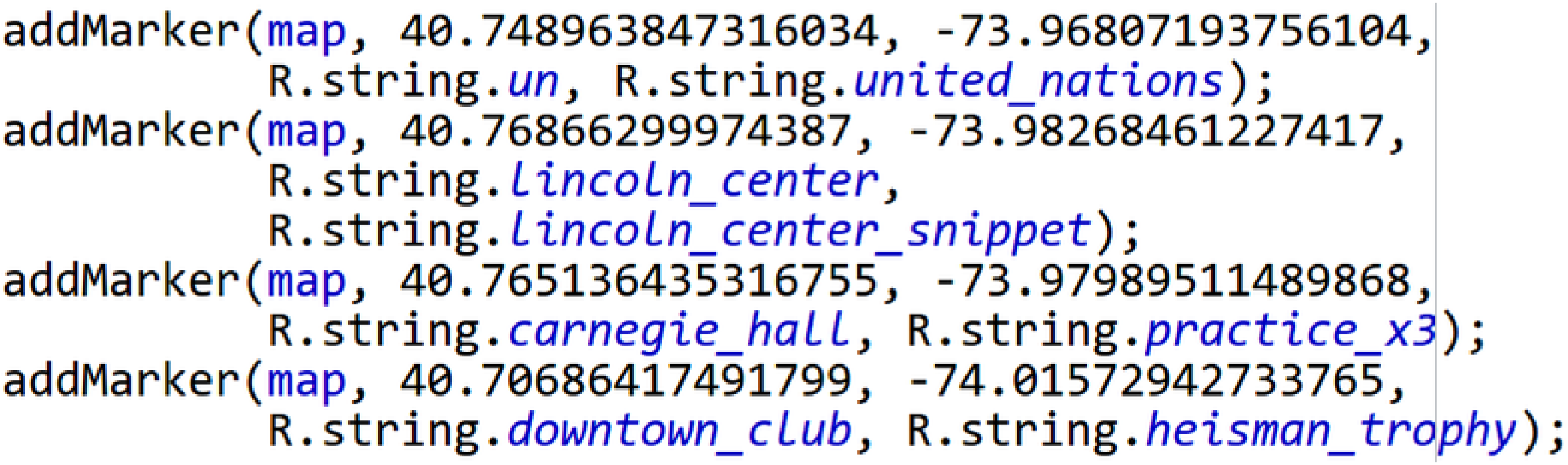
```
private void initListNav() {
  ArrayList<String> items=new ArrayList<String>();
  ArrayAdapter<String> nav=null;
  ActionBar bar=getSupportActionBar();

  for (int type : MAP_TYPE_NAMES) {
    items.add(getString(type));
  }

  if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.ICE_CREAM_SANDWICH) {
    nav=
        new ArrayAdapter<String>(
                                 bar.getThemedContext(),
                                 android.R.layout.simple_spinner_item,
                                 items);
  }
  else {
    nav=
        new ArrayAdapter<String>(
                                 this,
                                 android.R.layout.simple_spinner_item,
                                 items);
  }

  nav.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
  bar.setNavigationMode(ActionBar.NAVIGATION_MODE_LIST);
  bar.setListNavigationCallbacks(nav, this);
}
```



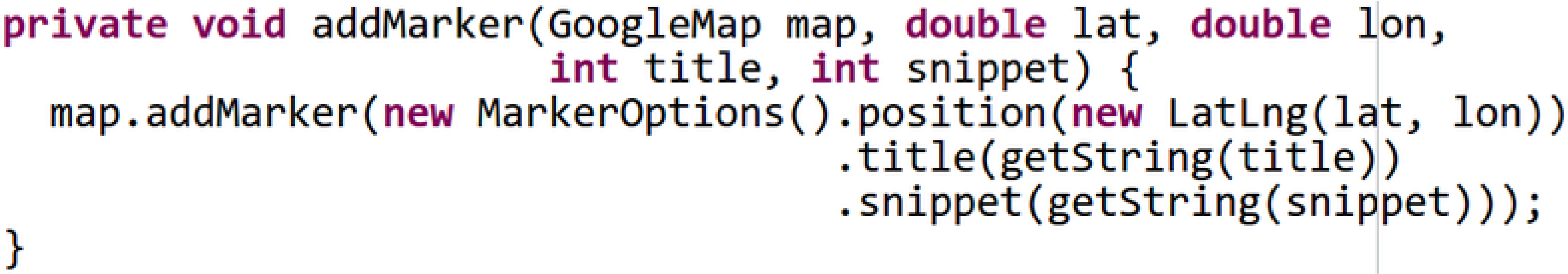
```
@Override
public boolean onNavigationItemSelected(int itemPosition,
  map.setMapType(MAP_TYPES[itemPosition]);

  return(true);
}
```

Marking Up a Map

- No More Overlays!
- Add Markers Via `addMarker()`
  - Takes a MarkerOptions object
  - Fluent API to describe marker
  - Position as a LatLng
    - No more microdegrees!
  - Provide title and snippet for default pop-up "info window"



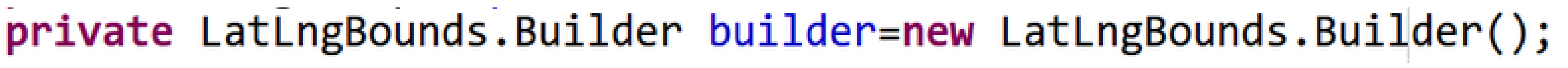
```
addMarker(map, 40.748963847316034, -73.96807193756104,
          R.string.un, R.string.united_nations);
addMarker(map, 40.76866299974387, -73.98268461227417,
          R.string.lincoln_center,
          R.string.lincoln_center_snippet);
addMarker(map, 40.765136435316755, -73.97989511489868,
          R.string.carnegie_hall, R.string.practice_x3);
addMarker(map, 40.70686417491799, -74.0157294273765,
          R.string.downtown_club, R.string.heisman_trophy);
```



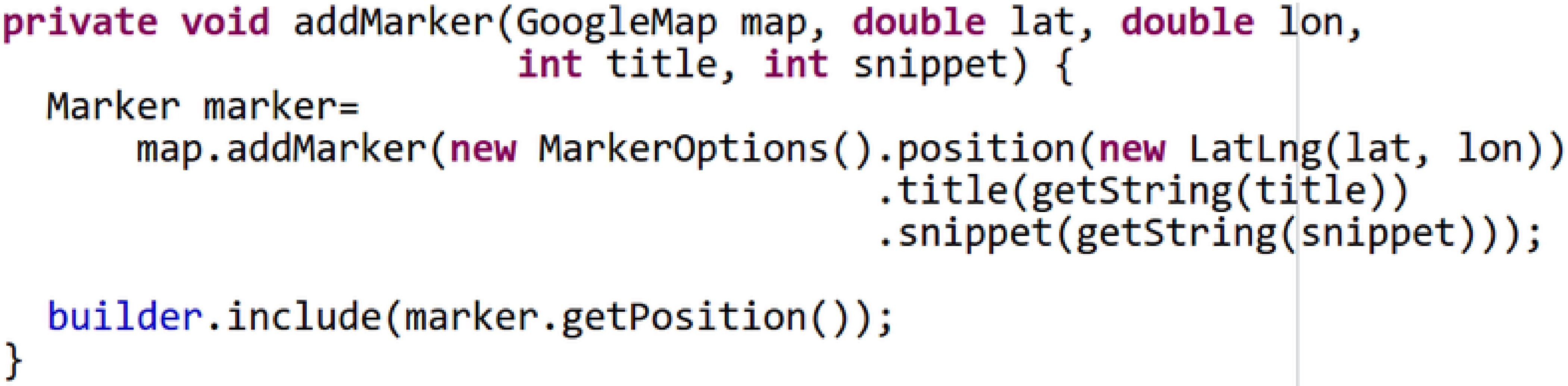
```java
private void addMarker(GoogleMap map, double lat, double lon,
                       int title, int snippet) {
  map.addMarker(new MarkerOptions().position(new LatLng(lat, lon))
                                   .title(getString(title))
                                   .snippet(getString(snippet)));
}
```

Seeing All the Markers

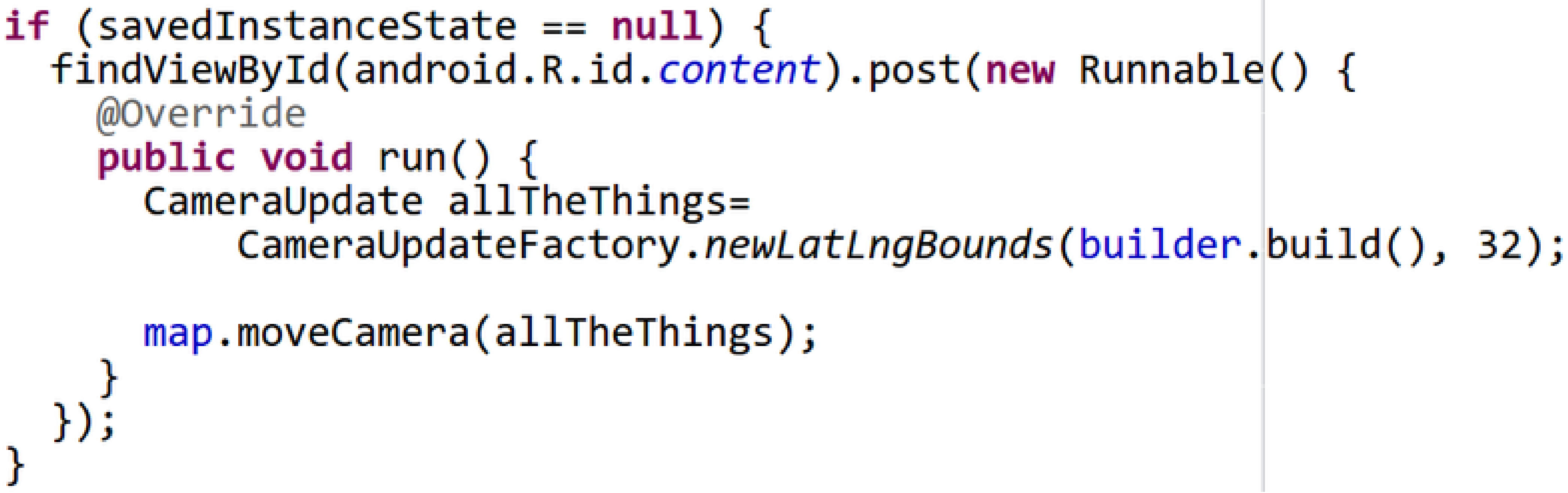
- Camera center, zoom may not show all of the markers that you add
  - Or may not show some specific subset that you really want the user to see from the outset
- Solution: LatLngBounds
  - Build one with LatLng locations for the markers in question
  - `CameraUpdateFactory.newLatLngBounds()`
  - `moveCamera()`



```
private LatLngBounds.Builder builder=new LatLngBounds.Builder();
```



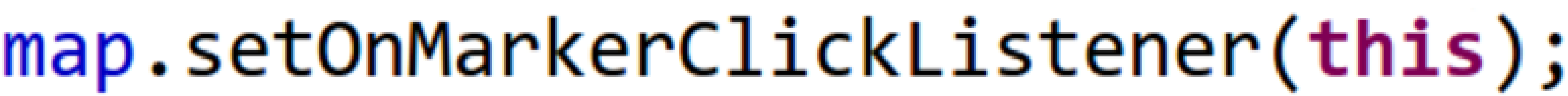
```java
private void addMarker(GoogleMap map, double lat, double lon,
                       int title, int snippet) {
  Marker marker=
      map.addMarker(new MarkerOptions().position(new LatLng(lat, lon))
                                       .title(getString(title))
                                       .snippet(getString(snippet)));

  builder.include(marker.getPosition());
}
```



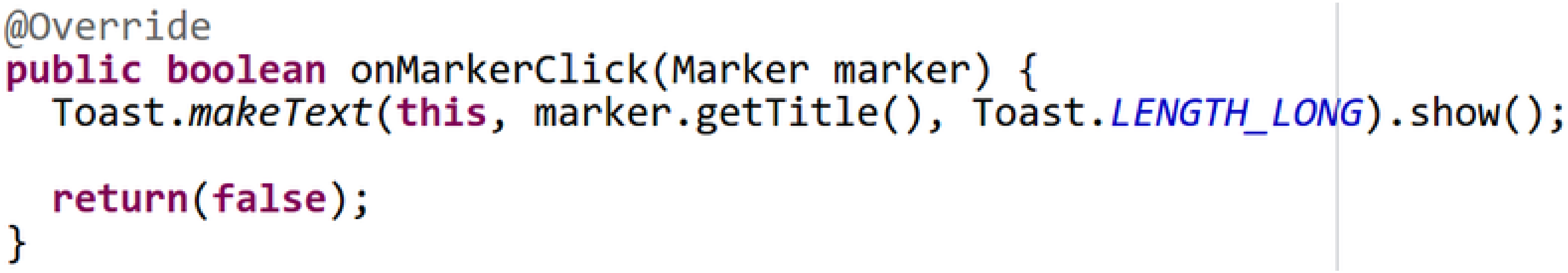
```
if (savedInstanceState == null) {
  findViewById(android.R.id.content).post(new Runnable() {
    @Override
    public void run() {
      CameraUpdate allTheThings=
          CameraUpdateFactory.newLatLngBounds(builder.build(), 32);

      map.moveCamera(allTheThings);
    }
  });
}
```

Taps, Sans Bugle

- Marker Taps
  - Implement OnMarkerClickListener interface
    - `onMarkerClick()` called when user taps on a Marker
  - Associate with GoogleMap via `setOnMarkerClickListener()`



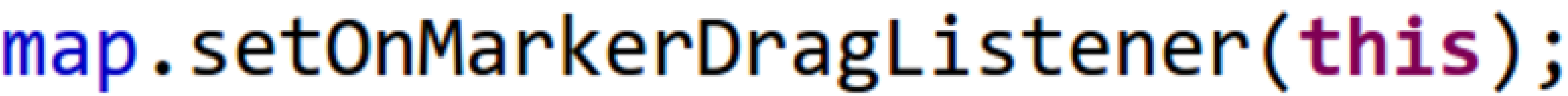
```
map.setOnMarkerClickListener(this);
```



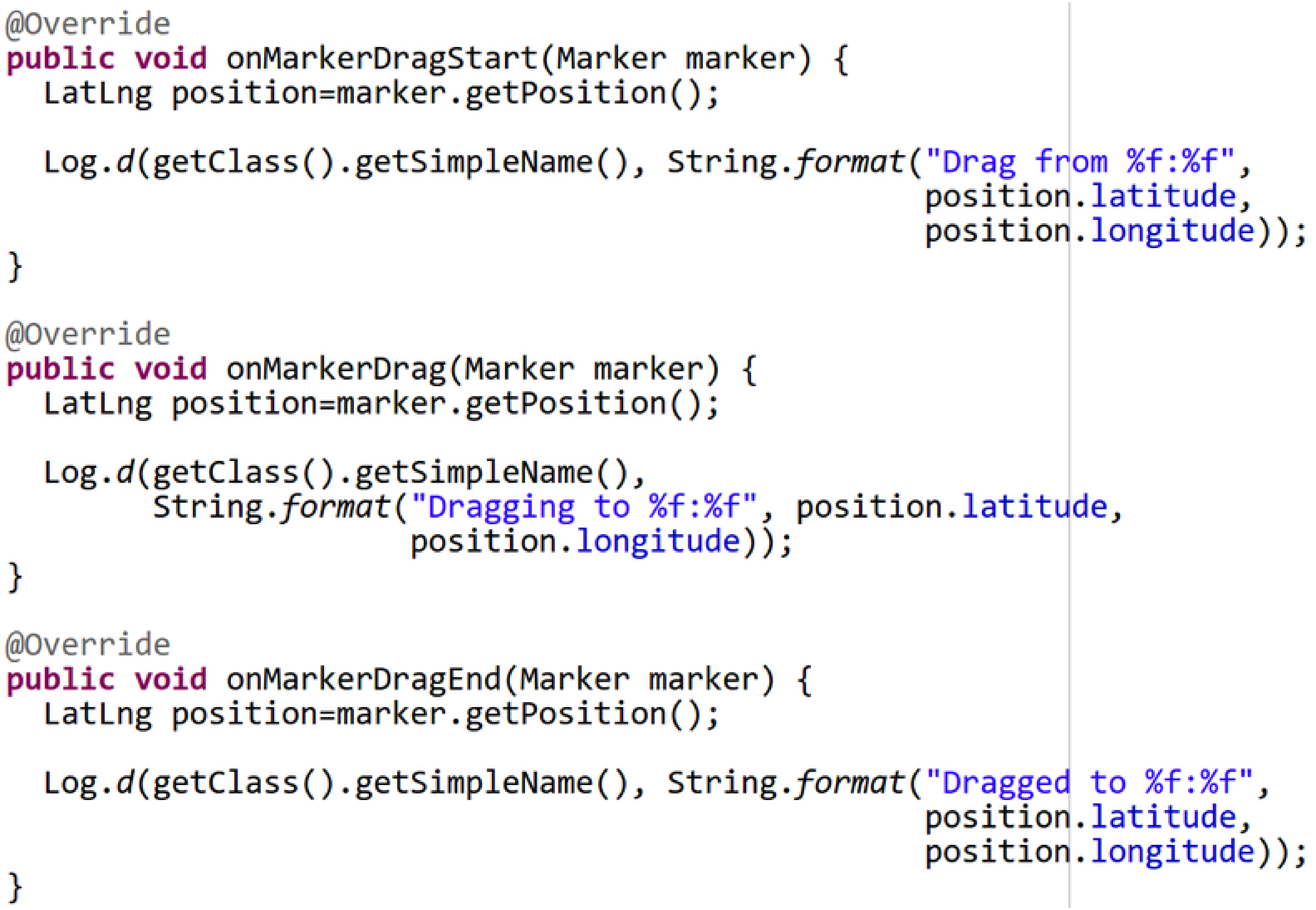
```
@Override
public boolean onMarkerClick(Marker marker) {
  Toast.makeText(this, marker.getTitle(), Toast.LENGTH_LONG).show();

  return(false);
}
```

Markers Can Be a Real Drag

- Marker Drag-and-Drop
  - Call `draggable(true)` on MarkerOptions
  - Implement OnMarkerDragListener interface
    - `onMarkerDragStart()`
    - `onMarkerDrag()`
    - `onMarkerDragEnd()`
  - Associate with GoogleMap via `setOnMarkerDragListener()`



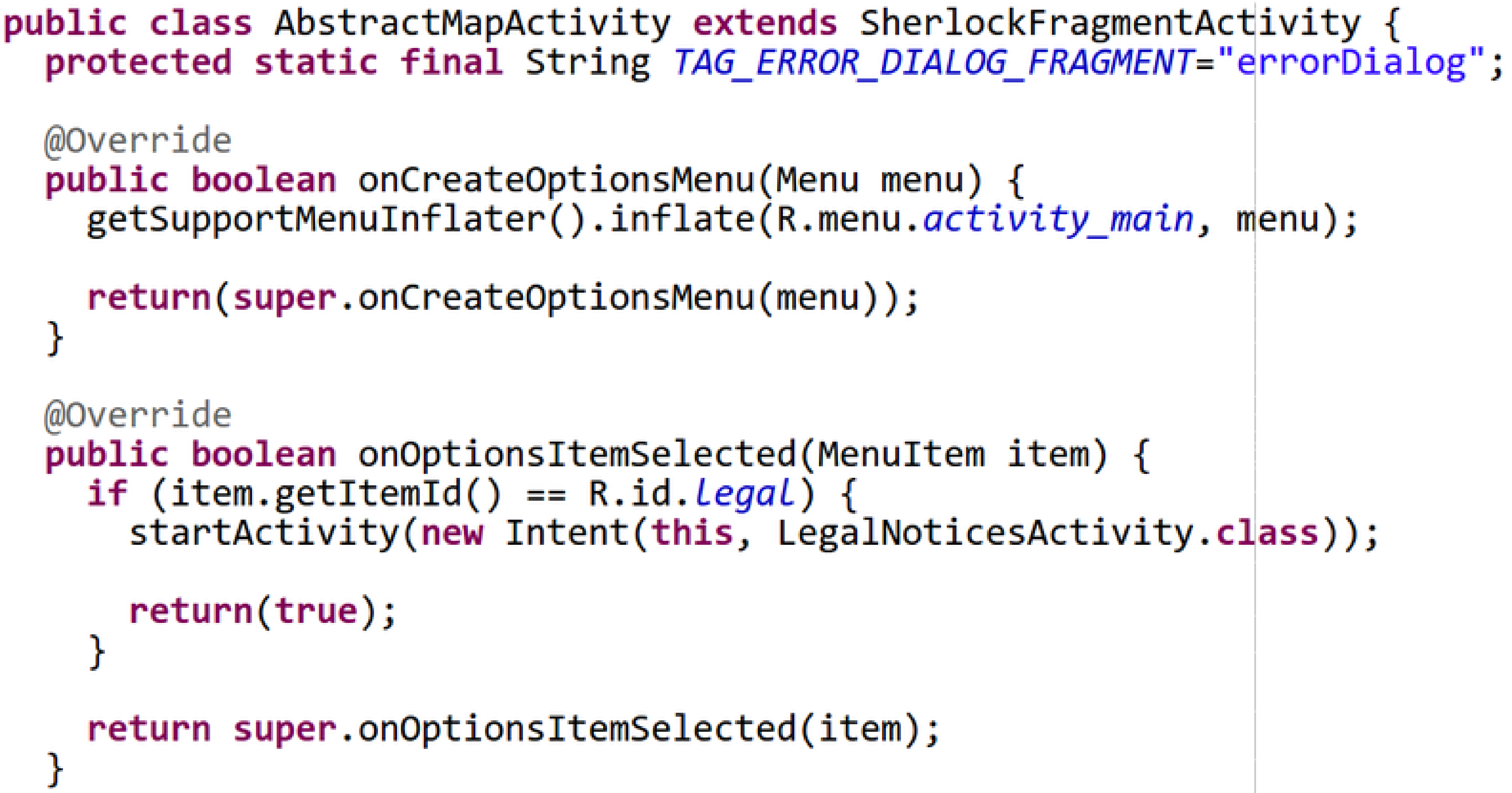
```
map.setOnMarkerDragListener(this);
```



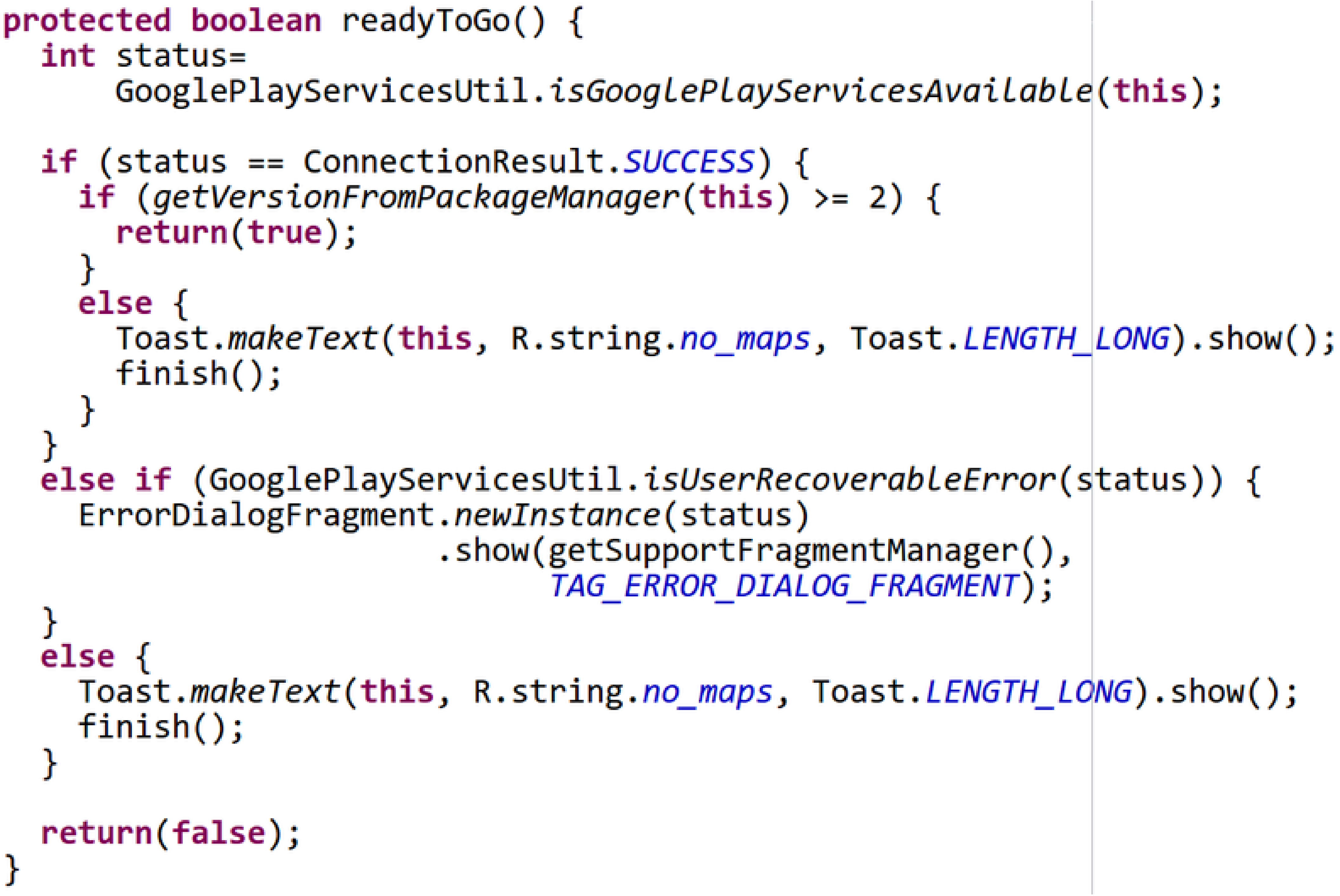
```
@Override
public void onMarkerDragStart(Marker marker) {
  LatLng position=marker.getPosition();

  Log.d(getClass().getSimpleName(), String.format("Drag from %f:%f",
                                                  position.latitude,
                                                  position.longitude));
}

@Override
public void onMarkerDrag(Marker marker) {
  LatLng position=marker.getPosition();

  Log.d(getClass().getSimpleName(),
        String.format("Dragging to %f:%f", position.latitude,
                      position.longitude));
}

@Override
public void onMarkerDragEnd(Marker marker) {
  LatLng position=marker.getPosition();

  Log.d(getClass().getSimpleName(), String.format("Dragged to %f:%f",
                                                  position.latitude,
                                                  position.longitude));
}
```

Dirty Details, Done Dirt Cheap

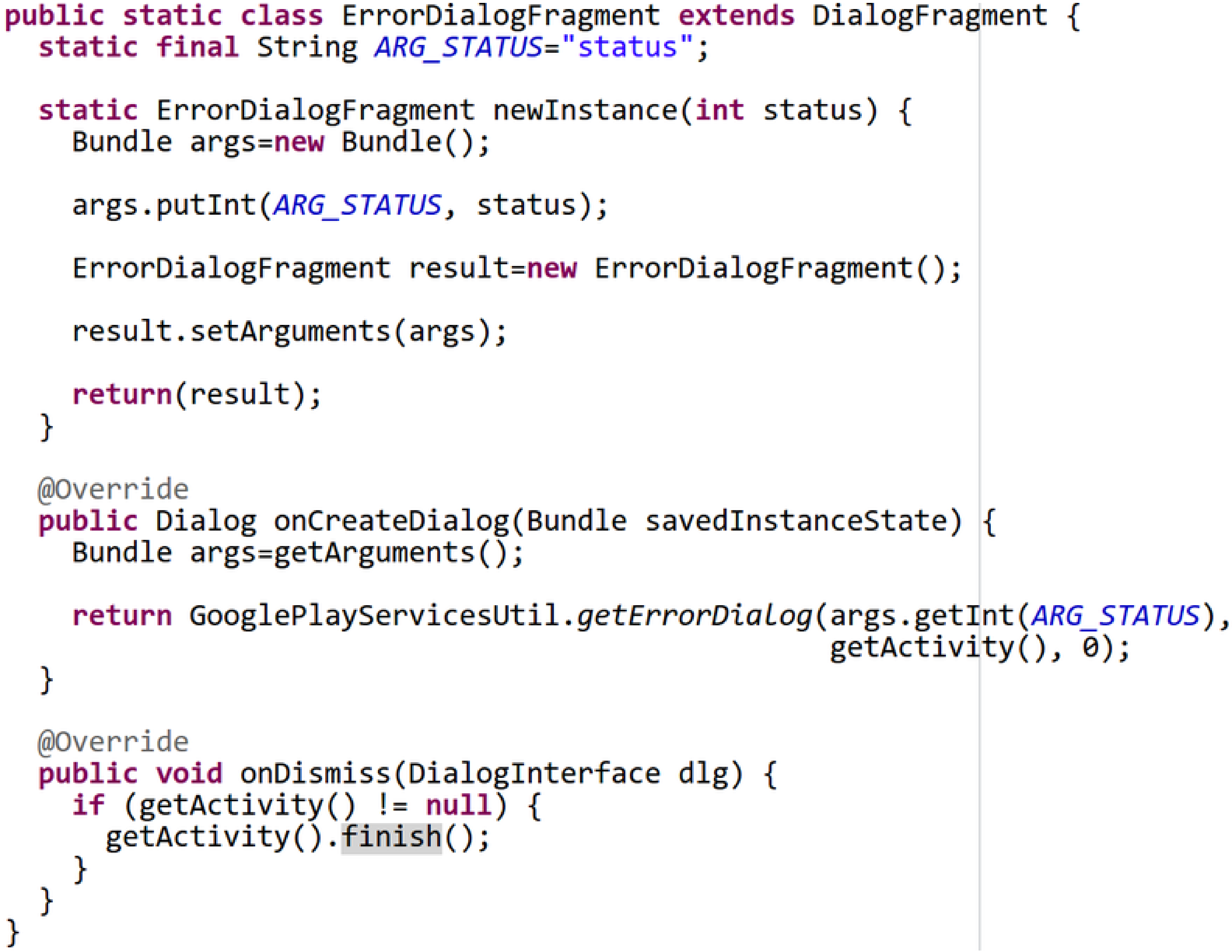
- AbstractMapActivity Features
  - Determining if Play Services Framework is available
  - Leading user to install Play Services Framework if possible
  - Showing license terms



```
public class AbstractMapActivity extends SherlockFragmentActivity {
  protected static final String TAG_ERROR_DIALOG_FRAGMENT="errorDialog";

  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    getSupportMenuInflater().inflate(R.menu.activity_main, menu);

    return(super.onCreateOptionsMenu(menu));
  }

  @Override
  public boolean onOptionsItemSelected(MenuItem item) {
    if (item.getItemId() == R.id.legal) {
      startActivity(new Intent(this, LegalNoticesActivity.class));

      return(true);
    }

    return super.onOptionsItemSelected(item);
  }
```



```java
protected boolean readyToGo() {
  int status=
      GooglePlayServicesUtil.isGooglePlayServicesAvailable(this);

  if (status == ConnectionResult.SUCCESS) {
    if (getVersionFromPackageManager(this) >= 2) {
      return(true);
    }
    else {
      Toast.makeText(this, R.string.no_maps, Toast.LENGTH_LONG).show();
      finish();
    }
  }
  else if (GooglePlayServicesUtil.isUserRecoverableError(status)) {
    ErrorDialogFragment.newInstance(status)
                       .show(getSupportFragmentManager(),
                             TAG_ERROR_DIALOG_FRAGMENT);
  }
  else {
    Toast.makeText(this, R.string.no_maps, Toast.LENGTH_LONG).show();
    finish();
  }

  return(false);
}
```



```
public static class ErrorDialogFragment extends DialogFragment {
  static final String ARG_STATUS="status";

  static ErrorDialogFragment newInstance(int status) {
    Bundle args=new Bundle();

    args.putInt(ARG_STATUS, status);

    ErrorDialogFragment result=new ErrorDialogFragment();

    result.setArguments(args);

    return(result);
  }

  @Override
  public Dialog onCreateDialog(Bundle savedInstanceState) {
    Bundle args=getArguments();

    return GooglePlayServicesUtil.getErrorDialog(args.getInt(ARG_STATUS),
                                                 getActivity(), 0);
  }

  @Override
  public void onDismiss(DialogInterface dlg) {
    if (getActivity() != null) {
      getActivity().finish();
    }
  }
}
```

But Wait! There's More!

- Advanced Maps V2
  - Info windows and custom contents
  - Drawing lines and shaded areas
  - Integrating location data
  - Animating map markers
  - And more!
  - http://goo.gl/UmxfPo

Slides! And Other Stuff Too!

http://commonsware.com/webinars/introMapsV2.html