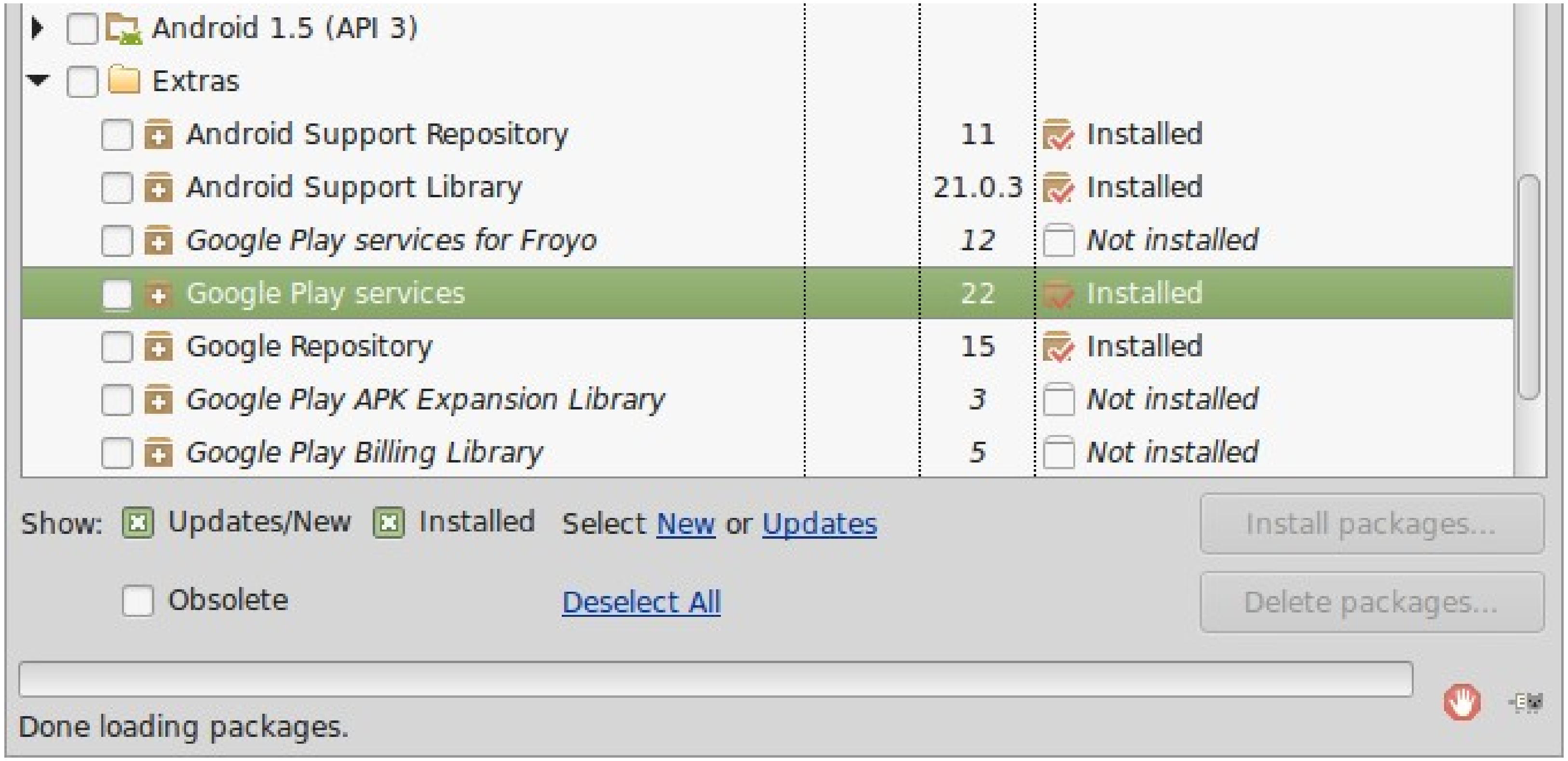
Maps and GPS

Installing Google Play services

- need to install **Google Play** services
  - SDK Manager → Extras → Google Play services (check box)
  - click Install packages...



Android 1.5 (API 3)		
Extras		
Android Support Repository	11	Installed
Android Support Library	21.0.3	Installed
Google Play services for Froyo	*12*	*Not installed*
Google Play services	22	Installed
Google Repository	15	Installed
Google Play APK Expansion Library	*3*	*Not installed*
Google Play Billing Library	*5*	*Not installed*

Show: ☒ Updates/New ☒ Installed Select New or Updates

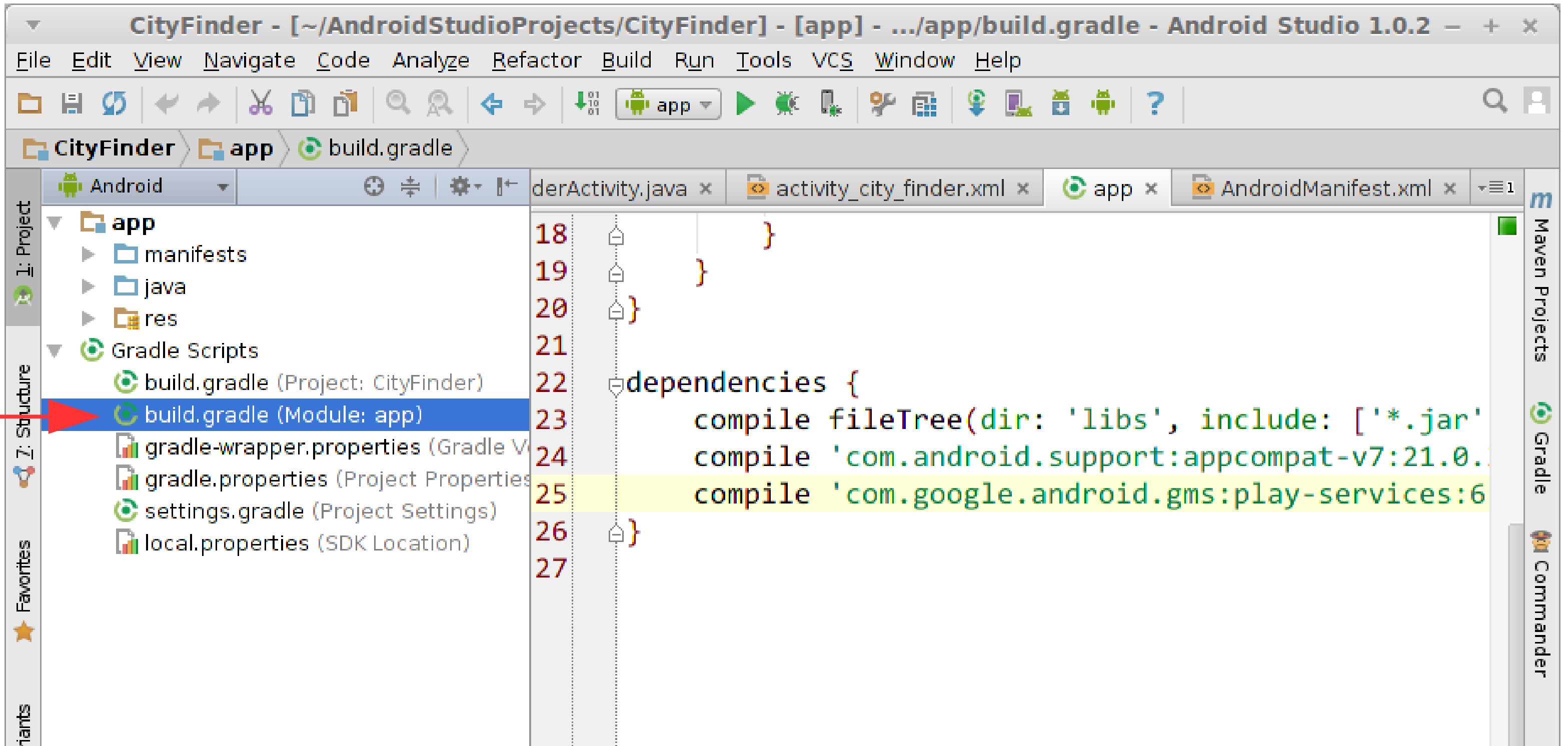
☐ Obsolete Deselect All

Done loading packages.

Adding Play Services to project

- add Google Play to project in app's **build.gradle** file

```
dependencies {
  compile fileTree(dir: 'libs', include: ['*.jar'])
  compile 'com.android.support:appcompat-v7:21.0.3'
  compile 'com.google.android.gms:play-services:6.5.87'
}
```


CityFinder - [~/AndroidStudioProjects/CityFinder] - [app] - .../app/build.gradle - Android Studio 1.0.2

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
CityFinder app build.gradle
Android
app
manifests
java
res
Gradle Scripts
build.gradle (Project: CityFinder)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle V
gradle.properties (Project Properties
settings.gradle (Project Settings)
local.properties (SDK Location)
derActivity.java
activity_city_finder.xml
app
AndroidManifest.xml
1: Project
7: Structure
Favorites
Maven Projects
Gradle
Commander

```
            }
        }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'
    compile 'com.android.support:appcompat-v7:21.0.
    compile 'com.google.android.gms:play-services:6
}
```

Get an API key, part 1

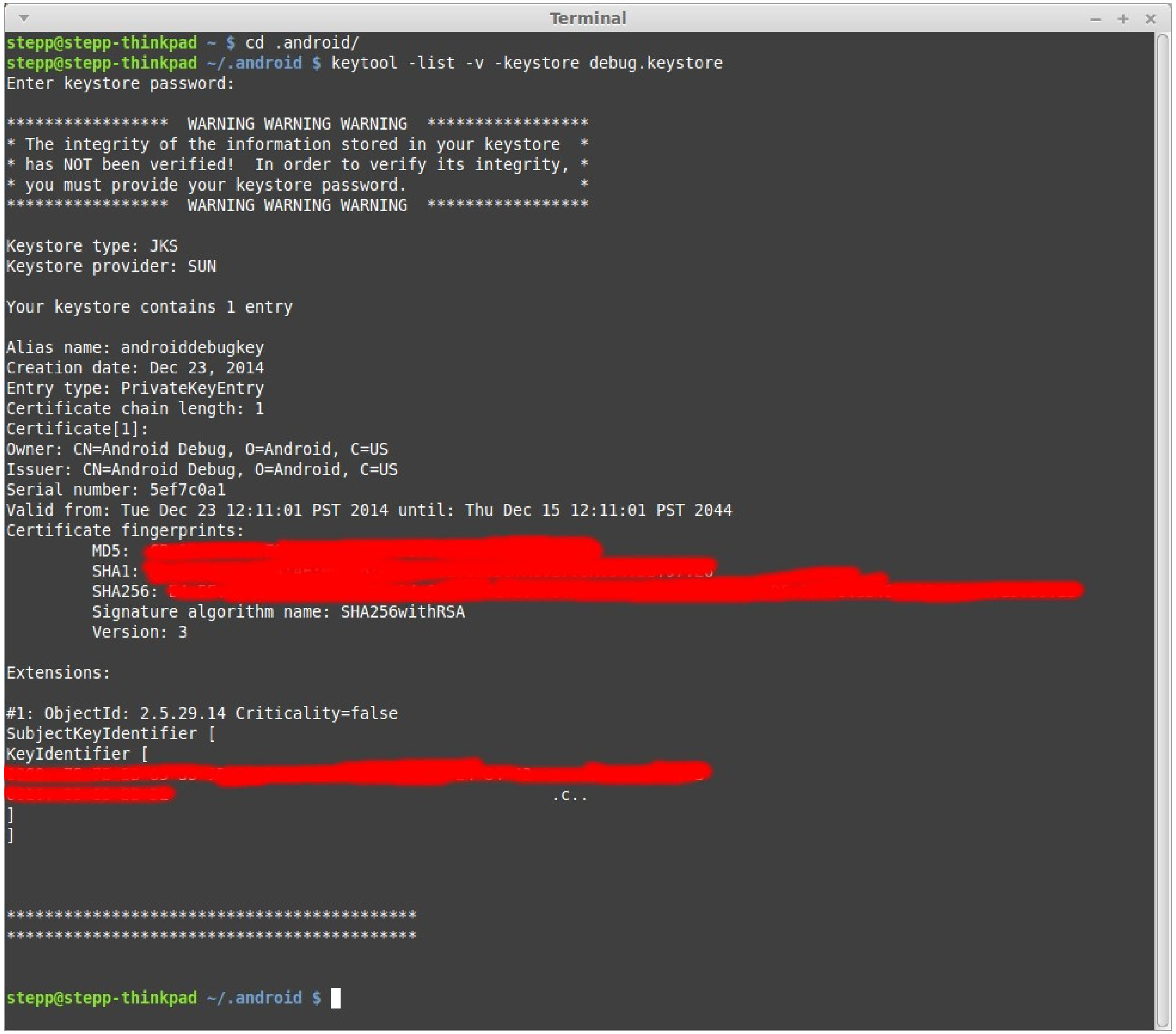
- Google won't allow you to fetch map data without an **API key**.
- To get a key, open a Terminal and find the file **debug.keystore**:
  - Windows (new): C:\Users\USERNAME\.android
  - Windows (old): C:\Documents and Settings\USERNAME\.android
  - Linux: /home/USERNAME/.android/
  - Mac: /Users/USERNAME/.android/ (?)
- In the terminal, **cd** to that directory, then type:

```
keytool -list -v -keystore debug.keystore
```

  (it asks for a password, so just press Enter)

- Find the line with your "Certificate fingerprint" for "SHA-1".
  It should contain a long string in this format. Copy it down.
  - BD:2B:3F:4B:.........

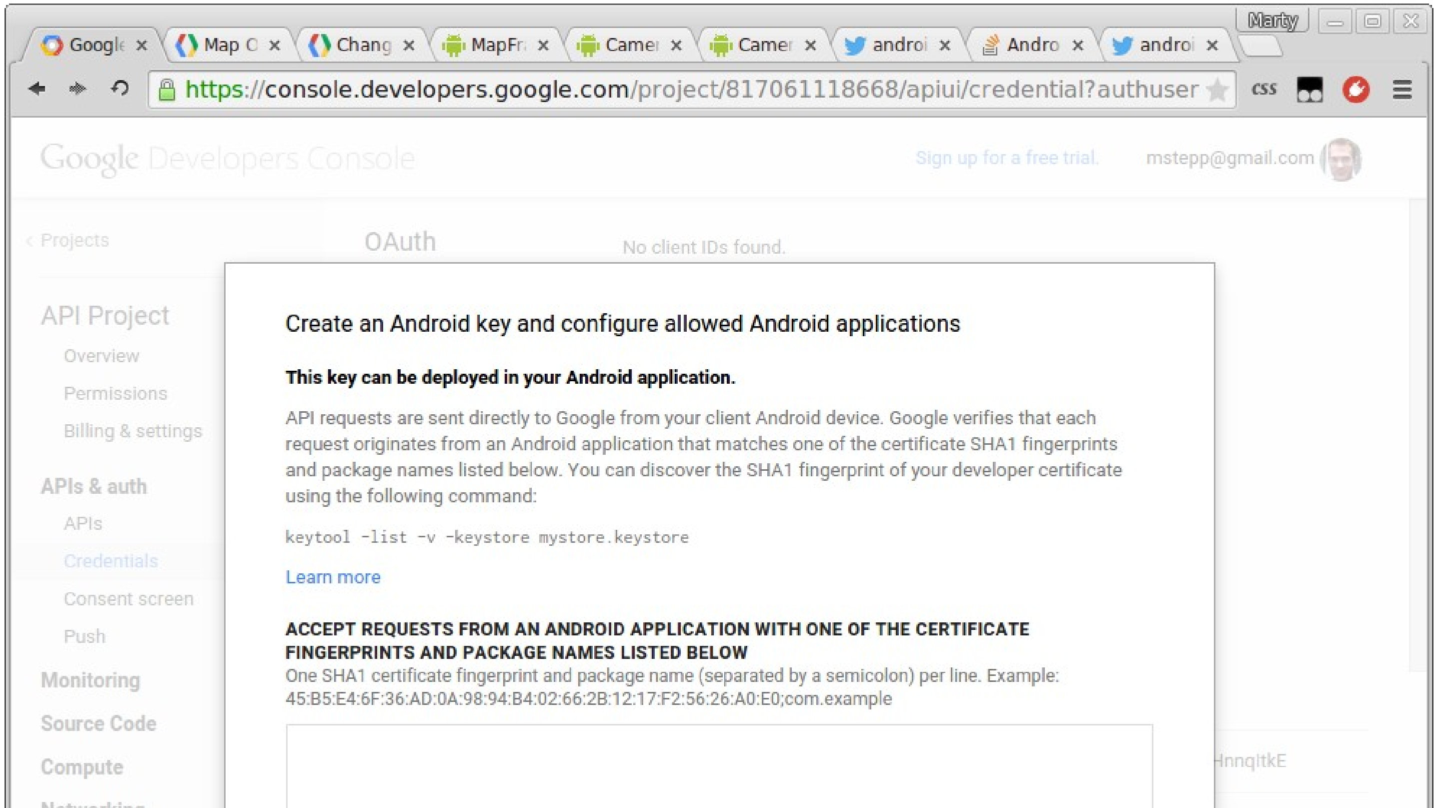
Get an API key, part 1 (screenshot)



```
stepp@stepp-thinkpad ~ $ cd .android/
stepp@stepp-thinkpad ~/.android $ keytool -list -v -keystore debug.keystore
Enter keystore password:

*****************  WARNING WARNING WARNING  *****************
* The integrity of the information stored in your keystore  *
* has NOT been verified!  In order to verify its integrity, *
* you must provide your keystore password.                  *
*****************  WARNING WARNING WARNING  *****************

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: androiddebugkey
Creation date: Dec 23, 2014
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 5ef7c0a1
Valid from: Tue Dec 23 12:11:01 PST 2014 until: Thu Dec 15 12:11:01 PST 2044
Certificate fingerprints:
         MD5:  [redacted]
         SHA1: [redacted]
         SHA256: [redacted]
         Signature algorithm name: SHA256withRSA
         Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
[redacted]
[redacted]                                                  .C..
]
]



*******************************************
*******************************************


stepp@stepp-thinkpad ~/.android $
```

Get an API key, part 2

- Go to the Google APIs developer console:
  - https://code.google.com/apis/console/
  - click APIs and Auth → Credentials → Create new Key
  - choose Android Key
  - paste in the SHA-1 key you got from the previous slide



Create an Android key and configure allowed Android applications

This key can be deployed in your Android application.

API requests are sent directly to Google from your client Android device. Google verifies that each request originates from an Android application that matches one of the certificate SHA1 fingerprints and package names listed below. You can discover the SHA1 fingerprint of your developer certificate using the following command:

```
keytool -list -v -keystore mystore.keystore
```

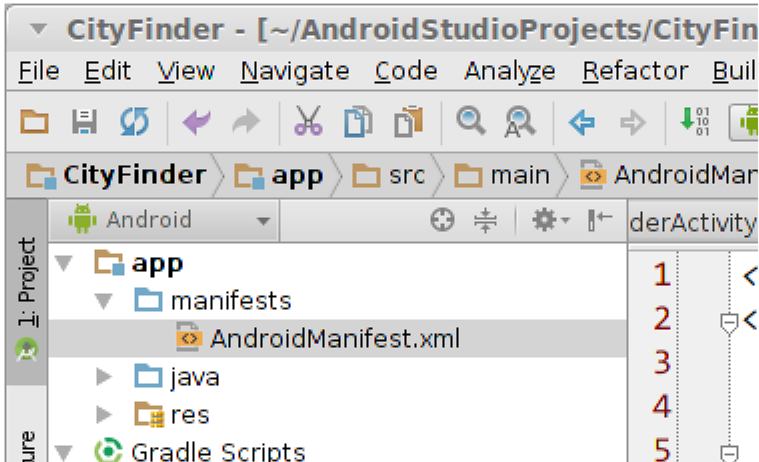
Learn more

ACCEPT REQUESTS FROM AN ANDROID APPLICATION WITH ONE OF THE CERTIFICATE FINGERPRINTS AND PACKAGE NAMES LISTED BELOW
One SHA1 certificate fingerprint and package name (separated by a semicolon) per line. Example:
45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A0:E0;com.example

AndroidManifest.xml changes

- To use maps in your app, must make some manifest changes:

```
<manifest ...>
   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
   <uses-permission android:name="android.permission.INTERNET" />
   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
   <uses-feature android:glEsVersion="0x00020000" android:required="true" />

   <application ...>
       <meta-data android:name="com.google.android.gms.version"
                  android:value="@integer/google_play_services_version" />
       <meta-data android:name="com.google.android.maps.v2.API_KEY"
                  android:value="your API key" />

       <activity ...> ... </activity>
   </application>
</manifest>
```

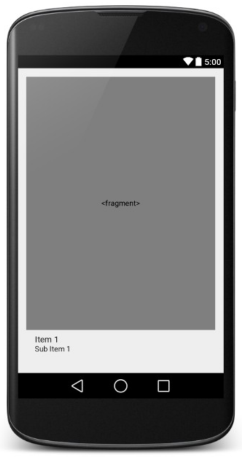

CityFinder - [~/AndroidStudioProjects/CityFin
File Edit View Navigate Code Analyze Refactor Buil
CityFinder app src main AndroidMar
Android
derActivity
1: Project
app
manifests
AndroidManifest.xml
java
res
Gradle Scripts

MapFragment (link)

- Google Maps API provides a fragment class named MapFragment for displaying a map within an activity.

```
<LinearLayout ...
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:map="http://schemas.android.com/apk/res-auto"
  tools:ignore="MissingPrefix">

  <fragment ...
      android:name="com.google.android.gms.maps.MapFragment"
      android:id="@+id/ID" />

</LinearLayout>
```

  - *(There is also a MapView class that we won't cover)*



Waiting for map to be ready

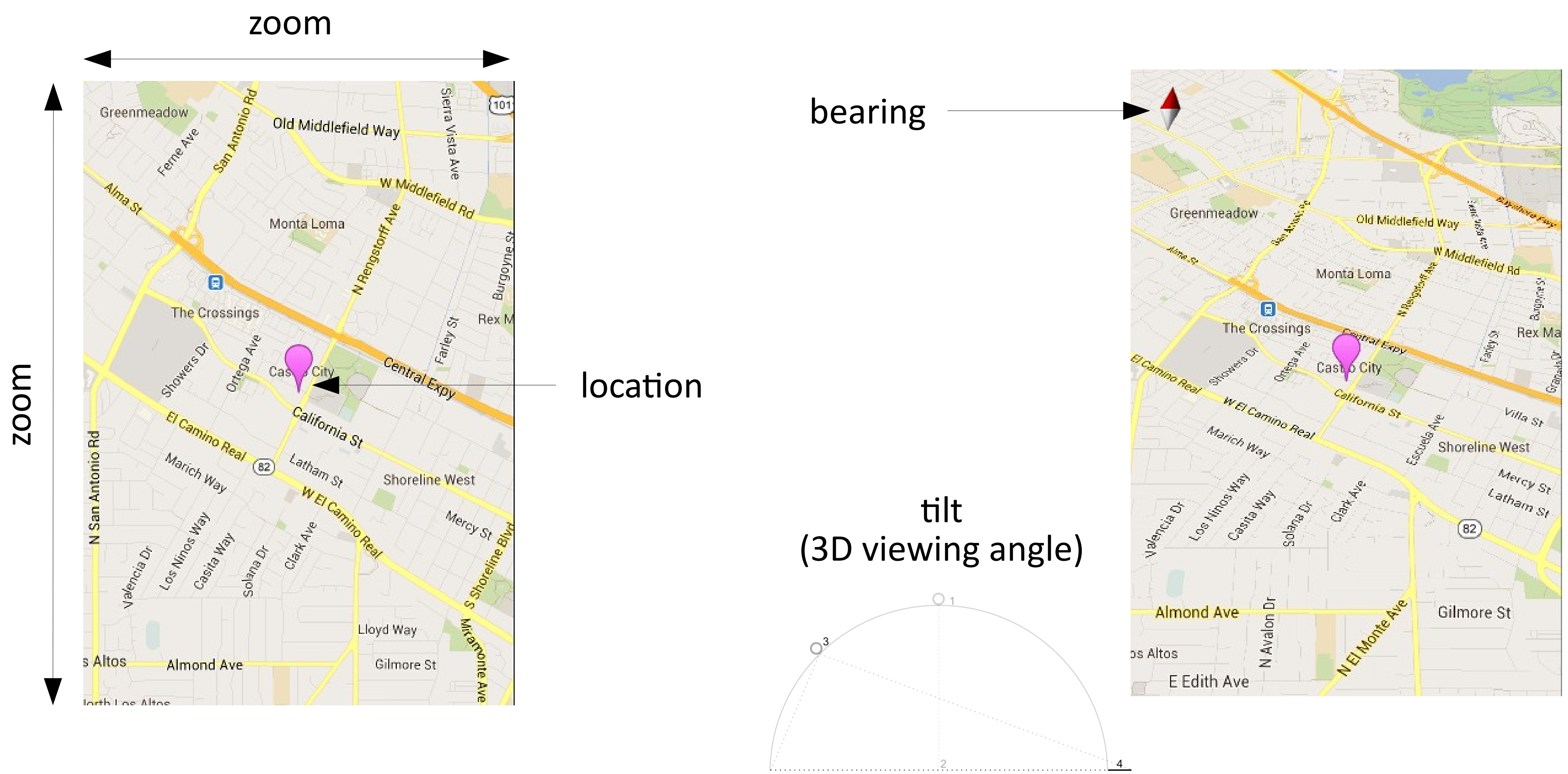
```
public class Name extends Activity
    implements OnMapReadyCallback, GoogleMap.OnMapLoadedCallback {
  private GoogleMap map = null;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    ...
    MapFragment mf = (MapFragment) getFragmentManager().findFragmentById(R.id.ID);
    mf.getMapAsync(this);                  // calls onMapReady when loaded
  }

  @Override
  public void onMapReady(GoogleMap map) {  // map is loaded but not laid out yet
    map.setOnMapLoadedCallback(this);      // calls onMapLoaded when layout done
  }

  @Override
  public void onMapLoaded() {
    code to run when the map has loaded;
  }
}
```

GoogleMap methods (link)

- placing items on the map:
  - addCircle, addGroundOverlay, **addMarker**, addPolygon, **addPolyline**, addTileOverlay
  - **clear** - Removes all markers, polylines/polygons, overlays
- manipulating the camera:
  - getCameraPosition, **moveCamera**, **animateCamera**, stopAnimation
- map settings and appearance:
  - setBuildingsEnabled, setIndoorEnabled, setMapType, setPadding, setTrafficEnabled
- snapshot - take a screen shot of the map as a bitmap
- event listeners:
  - setOnCameraChangeListener, **setOnMapClickListener**, setOnMapLoadedCallback, setOnMapLongClickListener, **setOnMarkerClickListener**, setOnMarkerDragListener, setOnMyLocationChangeListener

The map's camera

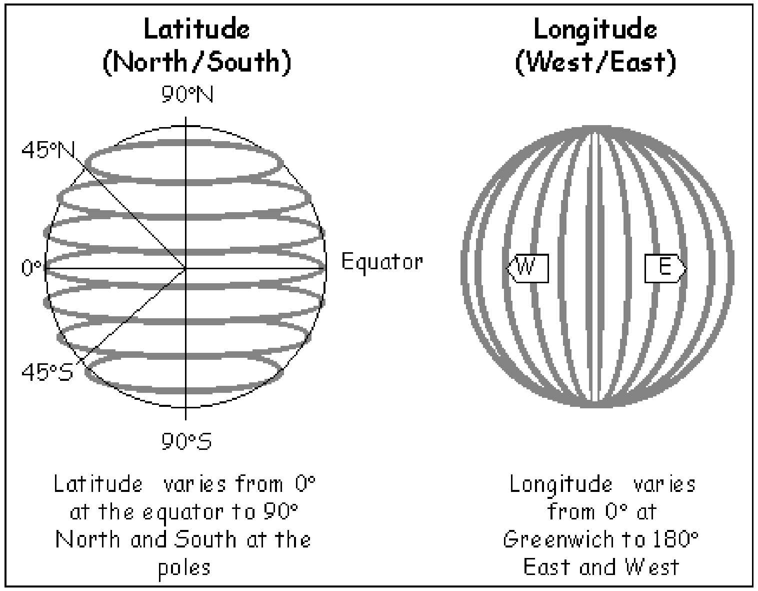
- The current viewing window of a map's camera is defined by:
  - **target** location (latitude/longitude), **zoom** (2.0 - 21.0),
  - **bearing** (orientation/rotation), and **tilt** (degrees)

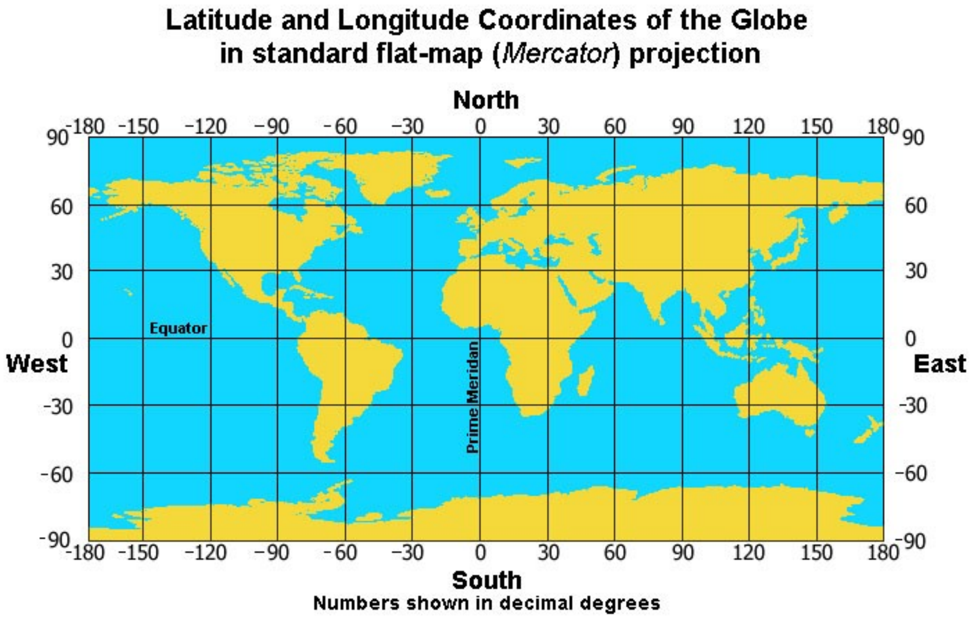


Latitude and longitude

- **latitude**: N/S angle relative to the equator
  - North pole = +90; South pole = -90
- **longitude**: E/W angle relative to prime meridian
  - West = 0 → -180;   East = 0 → 180
  - *find lat/lng of a place on Google Maps in URL address bar*

    see also: http://itouchmap.com/latlong.html

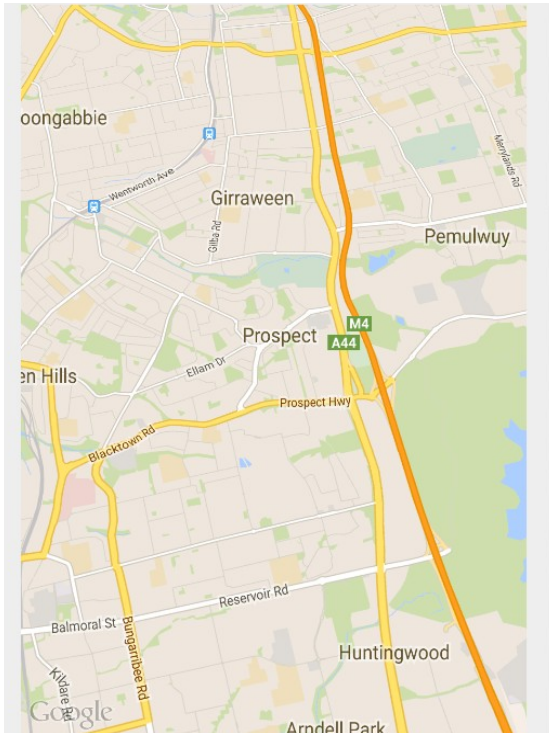

Latitude (North/South)
90°N
45°N
0°
Equator
45°S
90°S
Latitude varies from 0° at the equator to 90° North and South at the poles
Longitude (West/East)
W
E
Longitude varies from 0° at Greenwich to 180° East and West


Latitude and Longitude Coordinates of the Globe in standard flat-map (Mercator) projection
North
South
West
East
Equator
Prime Meridian
-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180
90 60 30 0 -30 -60 -90
Numbers shown in decimal degrees

Set camera in XML

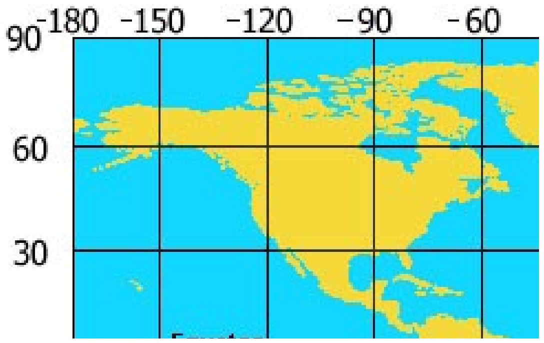
- Set initial map settings and camera position in the layout XML:
  - see here (link) for full list of attributes available

```
<fragment ...
    android:name="com.google.android.gms.maps.MapFragment"
    android:id="@+id/ID"
    map:cameraBearing="112.5"
    map:cameraTargetLat="-33.796923"
    map:cameraTargetLng="150.922433"
    map:cameraTilt="30"
    map:cameraZoom="13"
    map:mapType="normal"
    map:uiCompass="false"
    map:uiRotateGestures="true"
    map:uiScrollGestures="false"
    map:uiTiltGestures="true"
    map:uiZoomControls="false"
    map:uiZoomGestures="true" />
```



Set camera in Java code (link)

- CameraUpdateFactory methods:
  - newLatLng(new LatLng(***lat***, ***lng***))
  - newLatLngBounds(new LatLngBounds(***SW***, ***NE***), ***padding***)
  - newLatLngZoom(new LatLng(***lat***, ***lng***), ***zoom***)
  - newCameraPosition(***CameraPosition***)
  - others:

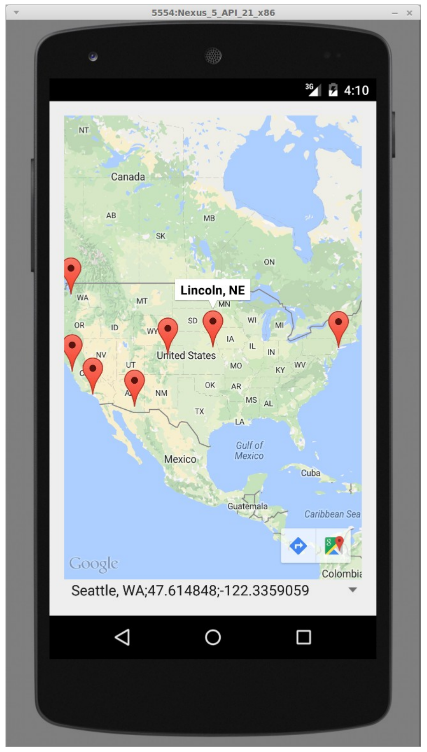


```
// example; show roughly the entire USA
LatLngBounds bounds = new LatLngBounds(
    new LatLng(20, -130.0),      // SW
    new LatLng(55,  -70.0));     // NE

map.moveCamera(CameraUpdateFactory.newLatLngBounds(bounds, 50));

// try also: map.animateCamera
```

Placing markers

- A `GoogleMap` object has an `addMarker` method that can let you put "push pin" markers at locations on the map.
  - The marker's methods return the marker, so you can chain them.
  - options: alpha, draggable, icon, position, rotation, title, visible, ...

```
map.addMarker(new MarkerOptions()
    .position(new LatLng(40.801, -96.691))
    .title("Lincoln, NE")
);

// to modify/remove the marker later
Marker mark = map.addMarker(new MarkerOptions()
    ...);
mark.remove();
```

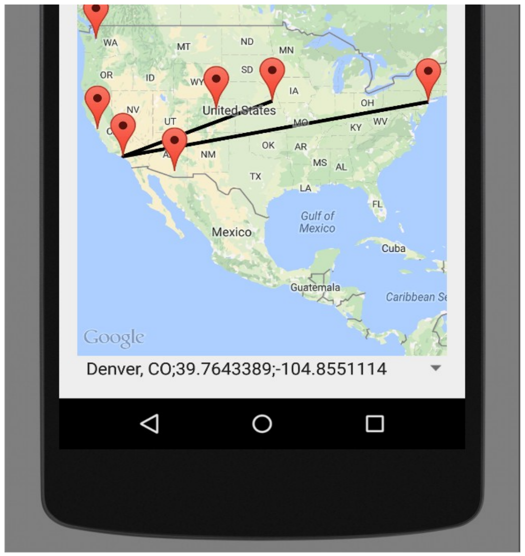


Seattle, WA;47.614848;-122.3359059

Lines and paths

- A GoogleMap object has an addPolyline method that can let you put lines between locations on the map.
  - options: color, visible, width, zIndex, ...

```
map.addPolyline(new PolylineOptions()
    .add(new LatLng(40.801, -96.691))    // Lincoln, NE
    .add(new LatLng(34.020, -118.412))   // Los Angeles, CA
    .add(new LatLng(40.703, -73.980))    // New York, NY
);

// to modify/remove the line later
Polyline polly = map.addPolyline(...);
polly.remove();
```



Denver, CO;39.7643389;-104.8551114

Accessing phone's location (link)

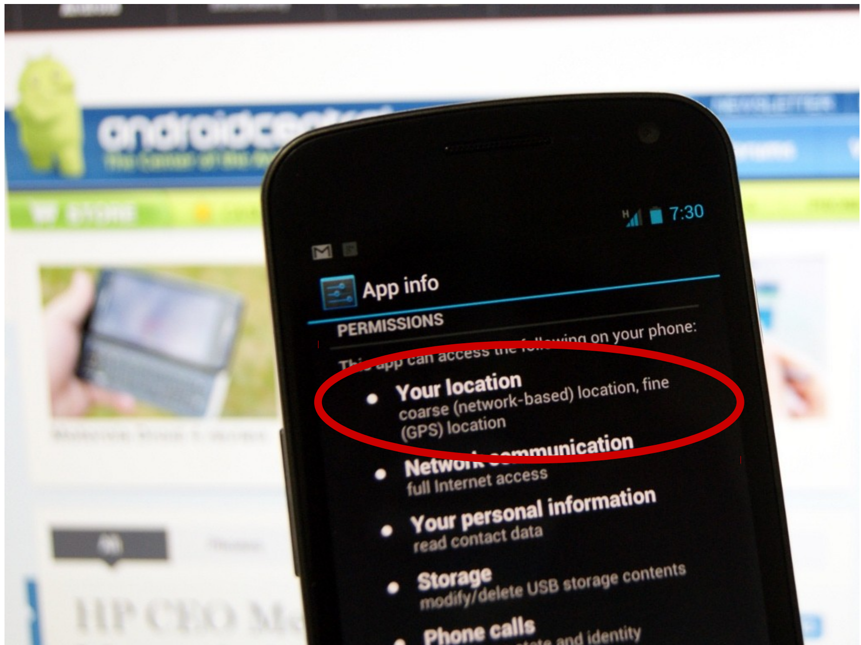
- Android `LocationManager` gives you the phone's position:
  - GPS provider provides highest accuracy
  - Network provider is a fallback in case GPS is disabled / not present

```
LocationManager locationManager = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);
Location loc = locationManager.getLastKnownLocation(
                    LocationManager.GPS_PROVIDER);
if (loc == null) {
    // fall back to network if GPS is not available
    loc = locationManager.getLastKnownLocation(
                    LocationManager.NETWORK_PROVIDER);
}
if (loc != null) {
    double myLat = loc.getLatitude();
    double myLng = loc.getLongitude();
    ...
    // other methods: getAltitude, getSpeed, getBearing, ...
```

AndroidManifest.xml changes

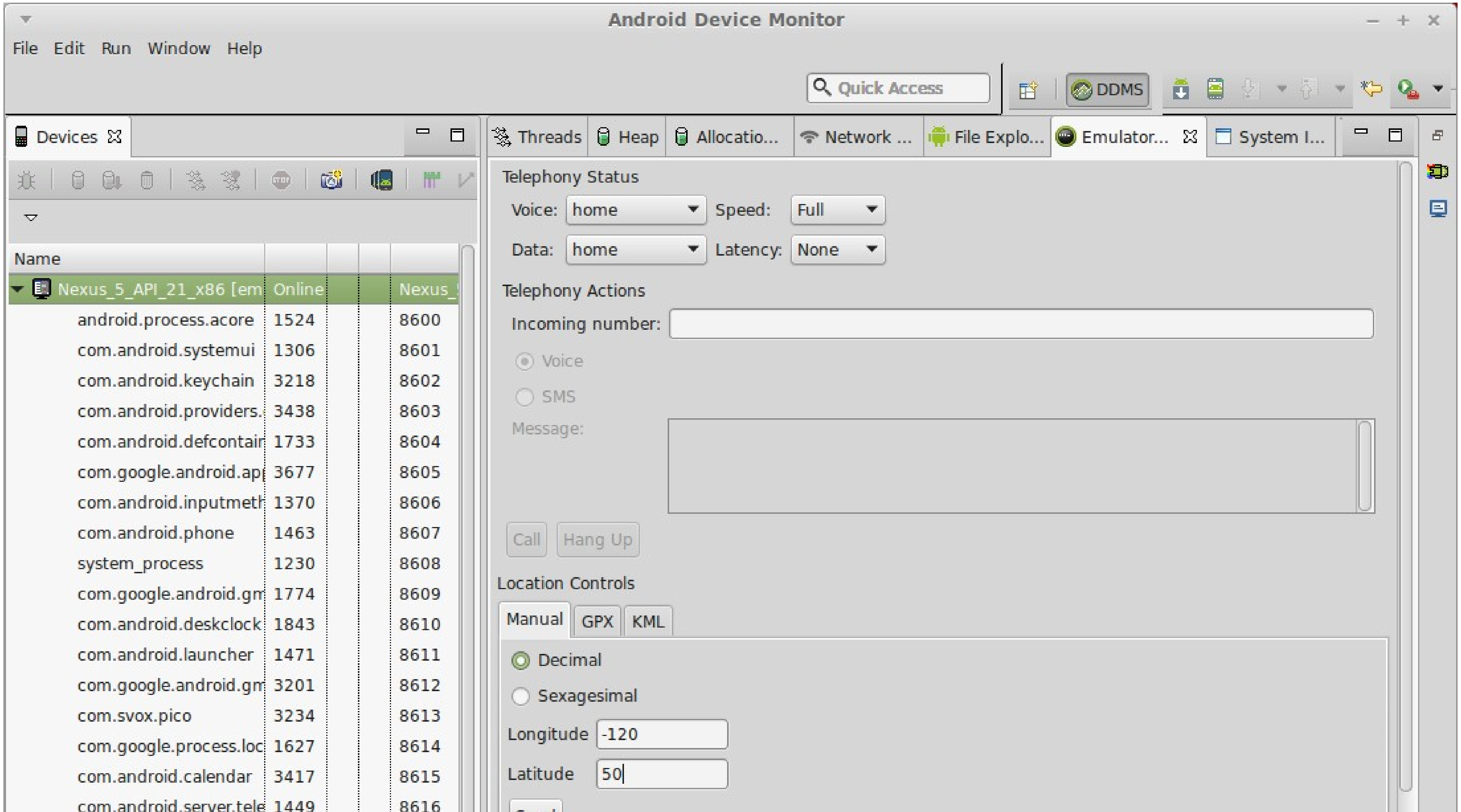
- Because of privacy issues, to access phone's current location, must request permission in `AndroidManifest.xml`:

```
<manifest ...>
   <uses-permission
       android:name="android.permission.ACCESS_COARSE_LOCATION" />
   <uses-permission
       android:name="android.permission.ACCESS_FINE_LOCATION" />

   <application ...>
       ...
   </application>
</manifest>
```


7:30
App info
PERMISSIONS
This app can access the following on your phone:
Your location
coarse (network-based) location, fine (GPS) location
Network communication
full Internet access
Your personal information
read contact data
Storage
modify/delete USB storage contents
Phone calls

Faking emulator's location (link)

- Android Device Monitor → Emulator Controls → Location
  - in device, click Settings → Location → On


Android Device Monitor
File Edit Run Window Help
Quick Access
DDMS
Devices
Threads Heap Allocatio... Network ... File Explo... Emulator... System I...

Name			
Nexus_5_API_21_x86 [em	Online		Nexus_
android.process.acore	1524		8600
com.android.systemui	1306		8601
com.android.keychain	3218		8602
com.android.providers.	3438		8603
com.android.defcontair	1733		8604
com.google.android.ap	3677		8605
com.android.inputmeth	1370		8606
com.android.phone	1463		8607
system_process	1230		8608
com.google.android.gm	1774		8609
com.android.deskclock	1843		8610
com.android.launcher	1471		8611
com.google.android.gm	3201		8612
com.svox.pico	3234		8613
com.google.process.loc	1627		8614
com.android.calendar	3417		8615
com.android.server.tele	1449		8616

Telephony Status

Voice: home Speed: Full

Data: home Latency: None

Telephony Actions

Incoming number:

Voice

SMS

Message:

Call Hang Up

Location Controls

Manual GPX KML

[✓ Decimal]

Sexagesimal

Longitude -120

Latitude 50

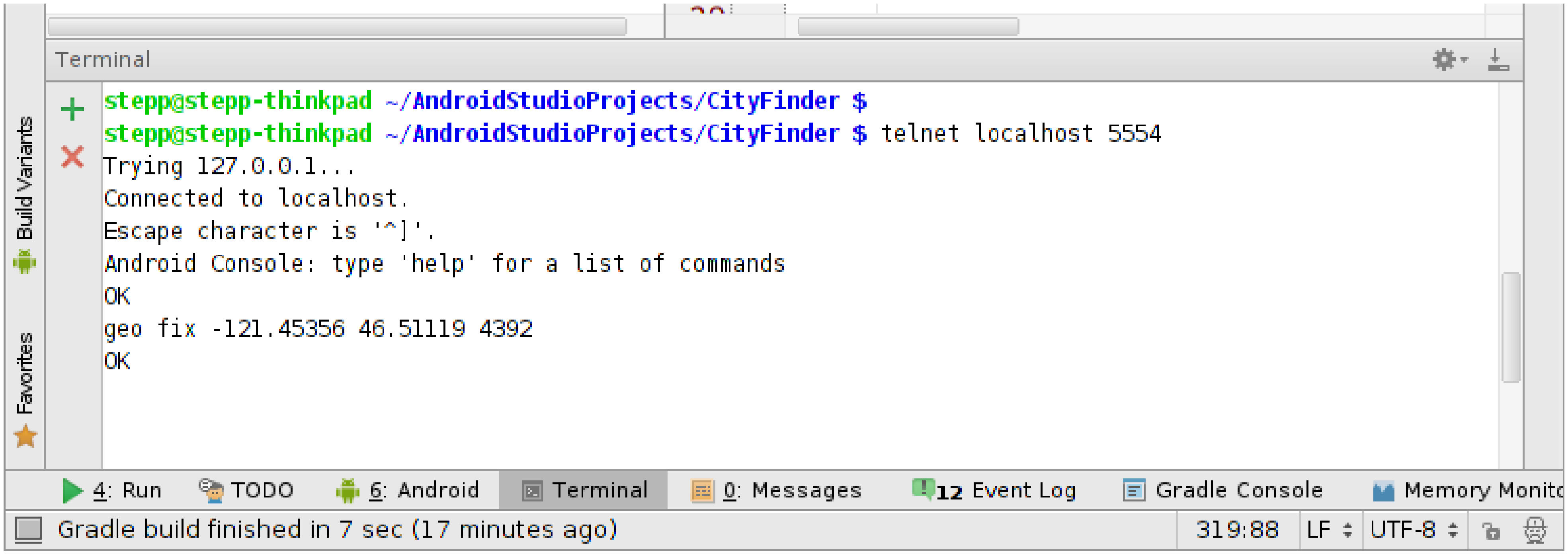
Faking emulator's location 2

- Another way: Open a **Terminal**, and type:

```
telnet localhost 5554
```

- once connected, type: *(altitude is optional)*

geo fix ***latitude longitude altitude***



```
stepp@stepp-thinkpad ~/AndroidStudioProjects/CityFinder $
stepp@stepp-thinkpad ~/AndroidStudioProjects/CityFinder $ telnet localhost 5554
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Android Console: type 'help' for a list of commands
OK
geo fix -121.45356 46.51119 4392
OK
```

Location update events

- Track user's movement by listening for location update events.

```
LocationManager locationManager = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);

locationManager.requestLocationUpdates(
    LocationManager.GPS_PROVIDER, 0, 0,   // provider, min time/distance
    new LocationListener() {
        public void onLocationChanged(Location location) {
            // code to run when user's location changes

        }
        public void onStatusChanged(String prov, int stat, Bundle b){}
        public void onProviderEnabled(String provider) {}
        public void onProviderDisabled(String provider) {}
    }
);
```