



<http://www.android.com/>

LBS intro

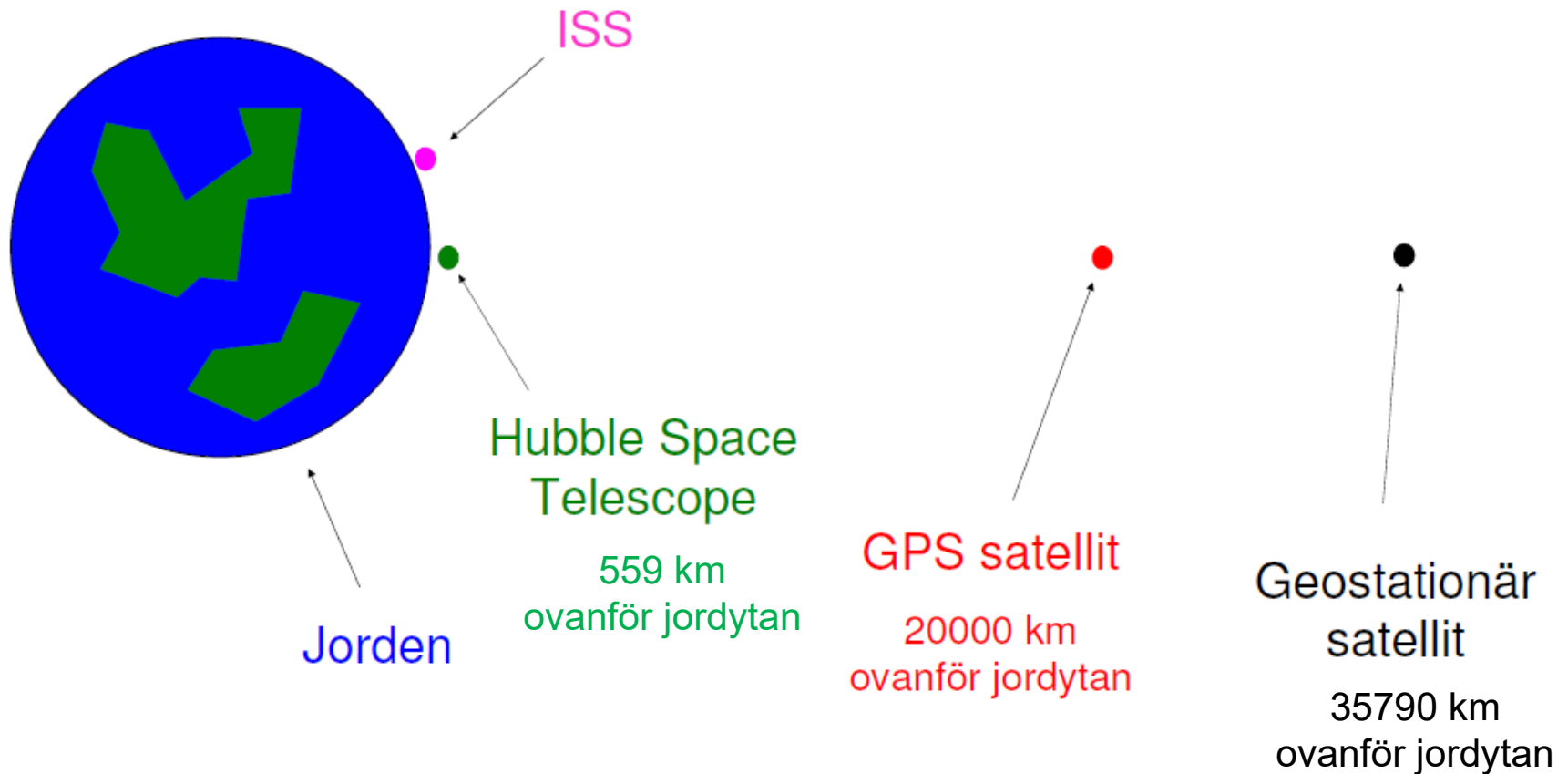
Location and map based services

Google APIs for Android / Play Services

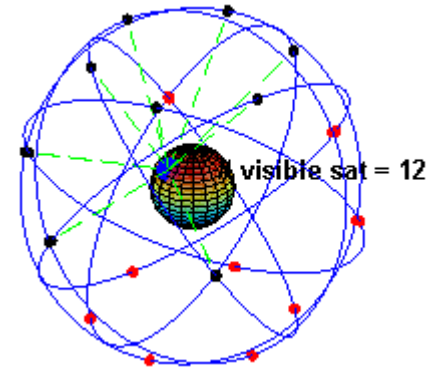
Sending objects references between activities or fragments

Satellitbaserade positioneringssystem

- http://sv.wikipedia.org/wiki/Geostation%C3%A4r_omloppsbanan



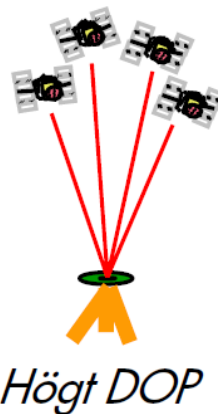
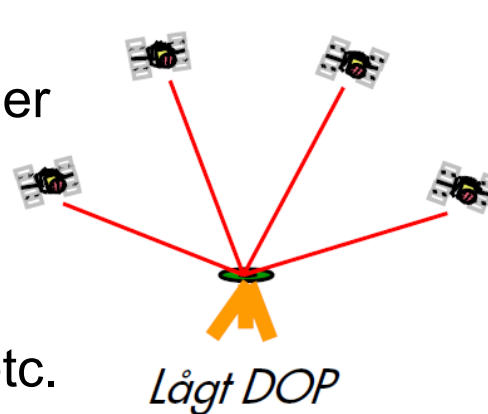
GPS



- Tänkt som ett militärt system från början, störnings signal borta 2000
- Första satelliten skickades upp 1978
- 1994 hade alla 24 satelliter samt tre reservsatelliter skickats upp
- För närvarande 24-32 satelliter
- Noggrannheten är ca: 10 m (stand alone)
- Inklination, 55 grader (ekvatorn = 0)
 - http://en.wikipedia.org/wiki/Global_Positioning_System
- Övriga satellitbaserade positionssystem
 - Glonass (Rus), Galileo (EU), Compass/BeiDou (Chn)

Fix och PDOP

- 1D (no fix), 2D or 3D Fix
 - At least 4 satellites is necessary to have a 3D position fix
- DOP = Mått på geometrisk spridning
 - Dilution Of Precision
 - HDOP, VDOP, PDOP, TDOP, GDOP, ...
 - [http://en.wikipedia.org/wiki/Dilution_of_precision_\(GPS\)](http://en.wikipedia.org/wiki/Dilution_of_precision_(GPS))
- Ju lägre DOP-tal-desto bättre geometrisk spridning
 - PDOP (position 3D) bör inte överstiga 4 - 5
- Elevationsmask bör vara 10-15 grader
 - 0 – 89 grader är teoretiskt
 - En lägre elevationsmask ger fler satelliter och ett bättre lägre DOP värde
 - Men ofta också ett högre "brus", reflektioner, atmosfär etc.



GPS devices & phones 1

- Built-in
 - Qualcomm gpsOne and other proprietary solutions
 - <http://www.prisjakt.nu/kategori.php?k=v615>
- Bluetooth (GPS chipsets)
 - U-blox, SiRF, RFMD and MTK etc.
- Static navigation filter
 - Bad for pedestrian mode
 - <http://etn.se/index.php?view=article&catid=27%3Aprodukt&id=47540%3Aantenn>
- Almanac
 - En förutsägelse om alla satelliternas bana och position runt jorden
 - Giltig uppemot flera månader
- Ephemeris
 - Innehåller enskilda satellitens exakta positionsdata vid given tid och annan information nödvändig för positionering, uppdateras var 30 sec.
 - Max uppemot 2-4h timmar gammal, 30 minuter i praktiken
 - http://www.satellitetrackingsystem.co.uk/satellite_tracking_software.htm



Sampling GPS, NMEA 0183

- The National Marine Electronics Association (NMEA)
 - http://en.wikipedia.org/wiki/NMEA_0183
- **GGA** - Essential fix data which provide 3D location and accuracy data
- **GSA** - GPS *DOP (dilution of precision) values and active satellites
- **GSV** - Satellites in view - 1 sentence data = max 4 satellites
- **RMC** - NMEA has its own version of essential GPS PVT (position, velocity, time) data
 - <http://www.gpsinformation.org/dale/nmea.htm#nmea>
- Draw a track on Google Map from a NMEA GPS log file
 - http://franson.com/forum/topic.asp?TOPIC_ID=4058 (GPS Visualizer)

- Example, 1 sec

\$GPGGA,133851.996,**6046.3885**,N,**01501.1250**,E,1,04,04.1,00172.3,M,30.0,M,,*63

\$GPGSA,A,3,06,30,,02,,05,,,,,,**07.3**,04.1,06.0*07

\$GPGSV,3,1,**11**,06,71,206,39,30,52,136,43,25,46,270,20,02,29,051,35*7B

\$GPGSV,3,2,**11**,23,15,340,30,05,22,127,35,21,12,181,35,10,11,093,*7B

\$GPGSV,3,3,**11**,16,11,291,,01,08,259,,33,01,251,*44

} Max 12 sat.
.

\$GPRMC,133851.996,A,6046.3885,N,01501.1250,E,064.7,194.4,**131105**,003.1,E*6^a

Projections, reference systems



- WGS 84 (World Geodetic System 1984)
 - Decimal Degrees
 - Lat/Long: 60.483494,15.411272
 - Degrees/Minutes/Seconds
 - Lat/Long: N 60° 29' 0.58", E 15° 24' 40.58"
 - Degrees/Decimalminutes
 - Lat/Long: 6029.3240,N,01523.3028,E
 - UTM (meters)
 - Cylindric map projection in 60 zones, SE=32-35
 - http://en.wikipedia.org/wiki/Transverse_Mercator_projection
 - <http://sv.wikipedia.org/wiki/UTM>
- RT90 - Swedish grid
 - X= 6707377 , Y= 1478345
- SWEREF 99 TM (använder UTM zon 33)
 - SWEdish REference Frame 1999
 - N: 6705329.393, E: 522604.151
- http://sv.wikipedia.org/wiki/WGS_84

Visa lat/long

- Decimalgrader
- Grader, minuter, sekunder
- Grader, decimalminuter
- Universal Transverse Mercator

To convert coordinates from degrees, minutes, seconds format to decimal format, use this formula:
 $\text{degrees} + (\text{minutes}/60) + (\text{seconds}/3600)$

To convert coordinates from degrees, decimal minutes format to decimal format, use this formula:
 $\text{degrees} + (\text{decimal minutes}/60)$

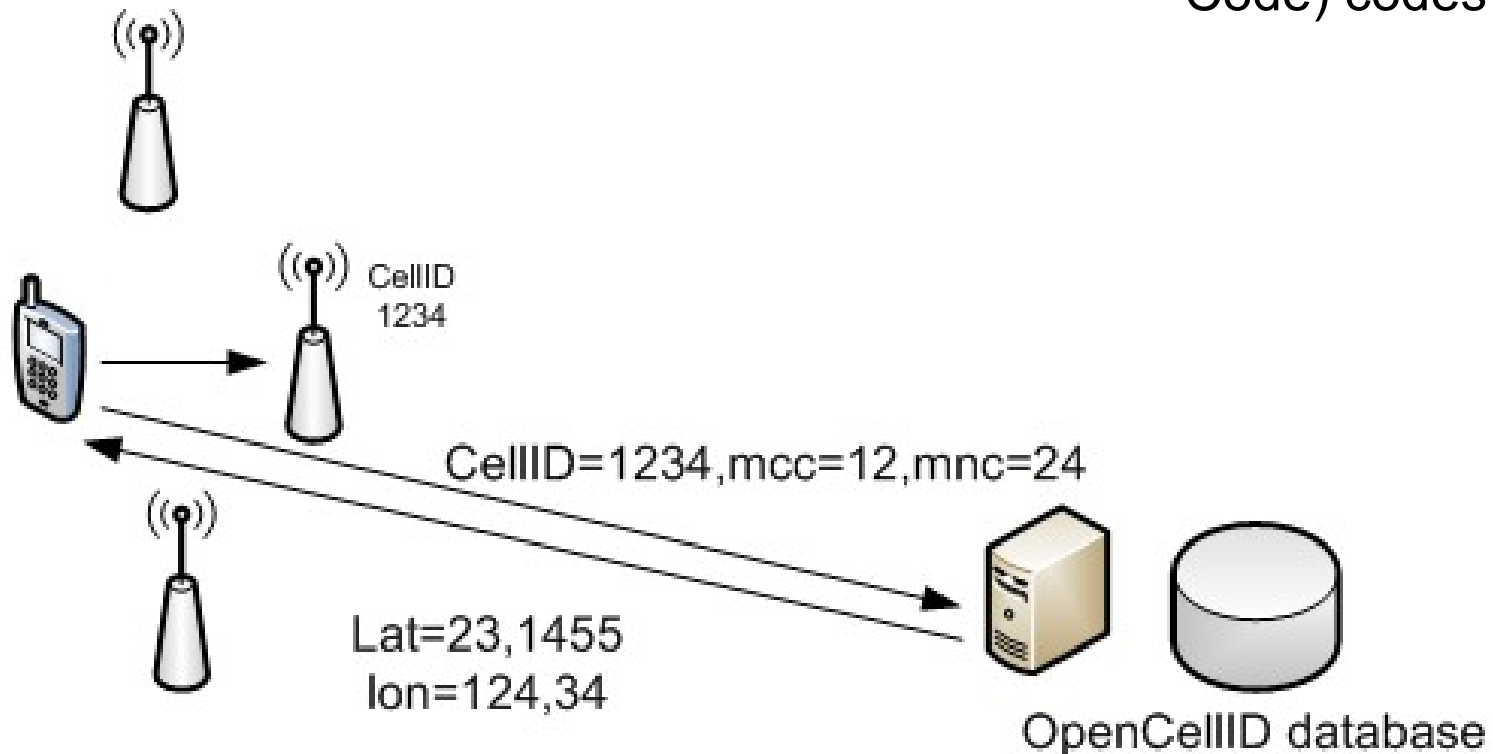
Mobila Positioneringsmetoder

- Mobilbaserad positionering
 - Mobilen tar emot signaler från positionssatta basstationer och positionerar sig med hjälp av informationen
 - WiFi nätverk etc. kan även användas
- Nätverksbaserad positionering (operatör)
 - Mobilnätverket positionerar den för det mesta passiva mobiltelefonen
- Mobilassisterad positionering
 - Mobilen gör alla mätningarna själv och skickar endast mätsignaler till nätverket för beräkning

CellID and WiFi positioning



- Basically both position modes works in a similar way
- WiFi uses the APs network cards hardware address (MAC) instead of the mobile networks Cell id, LAC (Location Area Code), MCC (Mobile Country Code) and MNC (Mobile Network Code) codes

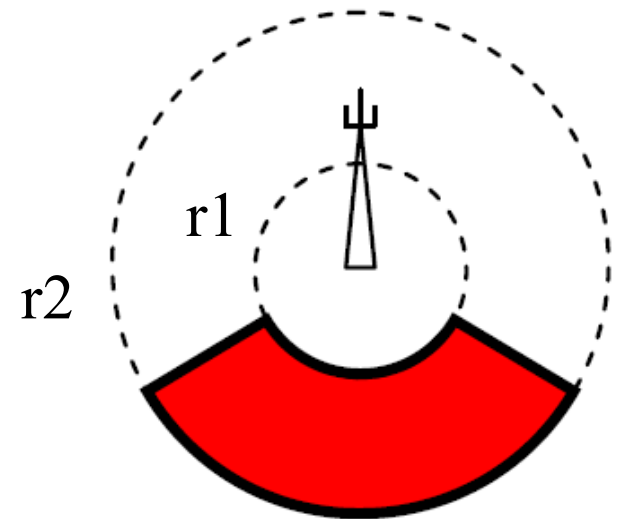


Cell ID



- Med Cell ID menas täckningsområdet av basstationens antenn/antennerna (sektorer)
 - Stort om den är rundstrålande
- CGI-TA
 - Cell Global Identity-Timing Advance
 - Del av sektor i basstationen vilka för det mesta strålar 120 grader
 - I GSM kan $r_2 - r_1$ som lägst bli 550 m, om inte cellen är mindre än 550 m
- Angel of Arrival (AOA)
 - Vinkel-radiell metod, kräver antenner som strålar mycket snävt

| Frequency (MHz) | Cell radius (km) | Cell area (km ²) | Relative cell count |
|-----------------|------------------|------------------------------|---------------------|
| 450 | 50 | 7521 | 1 |
| 950 | 26.9 | 2269 | 3 |
| 1800 | 14.0 | 618 | 12 |
| 2100 | 12 | 449 | 16 |



Mockup positions



- Via ADM or Extended controls as either GPX or KML files

The screenshot shows the Eclipse IDE with the DDMS (Dalvik Debug Monitor Service) interface. The main window displays the 'Location Controls' panel, which is currently set to 'GPX' mode. A table lists mockup locations with columns for Name, Longitude, Latitude, and Elevation. Below the table are playback controls including a play button, a 'Speed: 1X' dropdown, and a 'SEND' button.

The 'Extended controls' panel is also visible, showing various system settings like Location, Cellular, Battery, Phone, Directional pad, Fingerprint, Virtual sensors, Settings, and Help. The 'Location' section is expanded, showing 'GPS data point' settings (Coordinate system: Decimal, Longitude: -122.084) and 'Currently reported location' (Longitude: 15.3931, Latitude: 60.5021, Altitude: 171.0). The 'GPS data playback' section contains a table with columns for Delay (sec), Latitude, Longitude, Elevation, Name, and Description.

| Delay (sec) | Latitude | Longitude | Elevation | Name | Description |
|-------------|----------|-----------|-----------|------|-------------|
| 0 | 60.4981 | 15.3969 | 160 | | |
| 6 | 60.4982 | 15.3966 | 160 | | |
| 5 | 60.4984 | 15.3964 | 161 | | |
| 6 | 60.4986 | 15.396 | 162 | | |
| 5 | 60.4987 | 15.3958 | 162 | | |
| 5 | 60.4989 | 15.3955 | 163 | | |
| 5 | 60.499 | 15.3952 | 163 | | |

Location Based Services 1



- An application requires the following to access LBS from Android (the android.location API)
 - **LocationManager** - Class providing access to Android system location services, i.e. the Activity must be initialized with the Android LOCATION_SERVICE
 - **LocationListener** - Interface for receiving notifications from the LocationManager when the location has changed, a callback the Activity must implement
 - **Location** - Class representing a geographic location determined at a particular time, this is the result from the location service
- Different Criteras (android.location.Criteria) is possible to set as power requirements, position accuracy and provider etc.
- Make sure you remove position updates to save battery!
- Permissions in AndroidManifest

```
// GPS
<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION" />
// other positioning methods as WiFi (WLAN) and cell towers
<uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Location Based Services 2



- Get last location, use the `getLastKnownLocation(String provider)` method
- `minTime` and `minDistance` = `mininterval` (ms and meters) for notifications

```
private void initLocation() {
    // use the LocationManager class to obtain GPS locations
    LocationManager lm = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

    // providers: GPS_PROVIDER or NETWORK_PROVIDER
    Location loc = lm.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
    // params for requestLocationUpdates(String provider, long minTime,
    // float minDistance, LocationListener listener)
    lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, MyLocationListener);
}
// inner named class
private final LocationListener MyLocationListener = new LocationListener()
    @Override
    public void onLocationChanged(Location loc) {
        if (loc != null) {
            double lat = loc.getLatitude();
            double lng = loc.getLongitude();
            // do more ...
        }
    }
};
@Override
public void onPause() {
    lm.removeUpdates(MyLocationListener);
    super.onPause();
}

// good practice is to set paused to true
// before calling the super method
// mPaused is used to pause/resume work
@Override
protected void onPause() {
    mPaused = true;
    super.onPause();
}
// on resume set paused to false after
// calling the super method
@Override
protected void onResume() {
    super.onResume();
    mPaused = false;
}
```

Simple LocationManager



```
public class MyLocation
{
    LocationManager mLocationManager;
    Location mLocation;
    TextView tv;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        tv = (TextView) findViewById(R.id.TextView1);
        mLocationManager = (LocationManager)
            getSystemService(Context.LOCATION_SERVICE);

        Criteria criteria = new Criteria();
        criteria.setAccuracy(Criteria.ACCURACY_FINE);
        criteria.setPowerRequirement(Criteria.POWER_LOW);

        String locationprovider =
            mLocationManager.getBestProvider(criteria, true);
        mLocation =
            mLocationManager.getLastKnownLocation(locationprovider);
        tv.setText("Address: \nlatitude:" +
            mLocation.getLatitude() + "\nlongitude:" +
            mLocation.getLongitude());
    }
}
```

```
<uses-permission android:name="android.permission.INTERNET"/> // for Assisted-GPS?
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```



Geocoding 1



- You can translate an Address to a Location (Geocoding)

```
import java.util.List;
import java.util.Locale;
import android.location.Address;
import android.location.Geocoder;

// A class representing an Address, i.e, a set of Strings describing a location
List<Address> addresses;
String area = "Stockholm, Sweden"; // locationName
Geocoder gc = new Geocoder(this); // default system locale geocoder
try
{
    // Returns an array of Addresses that are known to describe the named location
    addresses = gc.getFromLocationName(area, 1); // maxResults = 1 is desired in this case
    if(addresses != null)
    {
        Address x = addresses.get(0);
        StringBuilder mSB = new StringBuilder("Address:\n");

        mSB.append("latitude: ").append(x.getLatitude()).append("\n")
            .append("longitude: ").append(x.getLongitude());
        Log.v(Constants.TAG, mSB.toString());
        mTV.setText(mSB.toString());
    }
    else
        Log.v(Constants.TAG, "addresses is null");
}
catch(IOException e){
    mTV.setText(e.getMessage());
}
```


Geocoding 2



- You can translate a Location to an Address (Reverse Geocoding)
- Emulator should work (especially if Google APIs system image)!

```
private void reverseGeoCode()
{
    List<Address> addresses;
    try {
        Geocoder mGC = new Geocoder(this, Locale.ENGLISH); // english locale geocoder
        // public List<Address> getFromLocation (double latitude, double longitude, int maxResults)
        addresses = mGC.getFromLocation(mLocation.getLatitude(),
            mLocation.getLongitude(), 1);
        if(addresses != null) {
            Address currentAddr = addresses.get(0);
            StringBuilder mSB = new StringBuilder("Address:\n");

            for(int i=0; i<currentAddr.getMaxAddressLineIndex(); i++)
                mSB.append(currentAddr.getAddressLine(i)).append("\n");

            mTV.setText(mSB.toString());
            Toast.makeText(getApplicationContext(), mSB.toString(),
                Toast.LENGTH_LONG).show();
        }
        else
            Log.v(Constants.TAG, "addresses is null");
    }
    catch(IOException e) {
        mTV.setText(e.getMessage());
    }
}
```

Using other sensors



- The Android SDK supports many different types of sensor devices (14+ sensors)
 - TYPE_ACCELEROMETER: Measures acceleration in the x-, y-, and z-axes
 - TYPE_LIGHT: Tells you how bright your surrounding area is
 - TYPE_MAGNETIC_FIELD: Returns magnetic attraction in the x-, y-, and z-axes
 - TYPE_GYROSCOPE: Measures the yaw, pitch, and roll of the device
 - TYPE_PRESSURE: Senses the current atmospheric pressure
 - TYPE_PROXIMITY: Provides the distance between the sensor and some object
 - TYPE_AMBIENT_TEMPERATURE: Measures the temperature of the surrounding area
 - TYPE_RELATIVE_HUMIDITY: Measures the humidity of the surrounding area
 - **Full list: <http://developer.android.com/reference/android/hardware/Sensor.html>**

```
private SensorManager mgr;
private final LinkedList<float[]> mFifo;
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mgr = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    mgr.registerListener(mySensorListener, mgr.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
        SensorManager.SENSOR_DELAY_GAME);
    // ...
}
// inner named class here, but can also be a whole class which implements SensorEventListener
private final SensorEventListener mySensorListener = new SensorEventListener() {
    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() != Sensor.TYPE_ACCELEROMETER)
            return;
        float[] val = new float[] { event.values[0], event.values[1],
            event.values[2] };
        mFifo.add(val);
        long nsTime = event.timestamp;
    }
}
```

Remember that do not spend too much time in callback methods or receivers!

Overview of Google Play Services



 Google APIs for Android

user base, and monetize more effectively with Google services.

HOME GUIDES REFERENCE

Search All Products

Develop

Ship higher quality apps. Faster.

Build a better app and leave the operations to us.

[Analytics](#)

[Cloud Messaging](#)

[Authentication](#)

[Realtime Database](#)

[Storage](#)

[Hosting](#)

[Remote Config](#)

[Test Lab](#)

[Sign in with Google](#)

[Maps](#)

[Places](#)

Grow

Grow an active user base

Acquire and engage the right users at the right time.

[Notifications](#)

[App Indexing](#)

[Dynamic Links](#)

[Invites](#)

[Adwords](#)

[App Install Ads](#)

Earn

Maximize your app revenue

Monetize your app through rich and engaging ads.

[AdMob](#)

<https://developers.google.com/android/>

<https://code.tutsplus.com/tutorials/search/Play+Services>

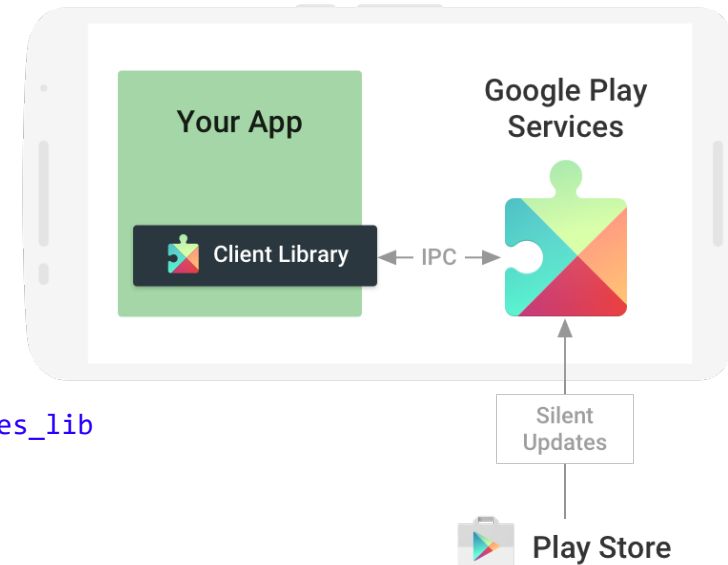
Using Google Maps API V2



- To use the the Google Maps API V2
 - Your AVD and project must include the Google Play Services library
 - Android Studio

```
dependencies {  
    compile 'com.google.android.gms:play-services:9.4.0'  
}
```

- Eclipse
 - Project properties > Android > Library or Project properties > Java Build Path > Libraries tab > Android Dependencies
 - In the project.properties file
`android.library.reference.1=../google-play-services_lib`
- Obtain a valid API key to use the with the Maps web service from
 - <https://console.developers.google.com>
- Include a MapFragment in the Activity layout XML file as below



```
<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.MapFragment" />
```

Using Google Maps API V2



- Set up the AndroidManifest.xml with Google Maps API V2
 - Add at least the following meta data, permissions and uses-* directives (more if GPS etc. is to be used)

```
<manifest>
  <application
    ...
    <meta-data
      android:name="com.google.android.geo.API_KEY"
      android:value="your_maps_key_here" />
    <meta-data
      android:name="com.google.android.gms.version"
      android:value="@integer/google_play_services_version" />
  </application>

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

  <permission
    android:name="du.se.mapsproject.permission.MAPS_RECEIVE"
    android:protectionLevel="signature"/>

  <uses-permission android:name="du.se.mapsproject.permission.MAPS_RECEIVE"/>
  <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
  <uses-permission android:name="android.permission.INTERNET"/>
</manifest>
```

```
/* adding the full library may give problems, add less! */
dependencies {
  compile 'com.google.android.gms:play-services-maps:9.4.0'
  compile 'com.google.android.gms:play-services-location:9.4.0'
}
```

Using Google Maps API V2



```
public class MapsActivity extends AppCompatActivity implements OnMapReadyCallback {
    private GoogleMap mMap; // might be null if Google Play services APK is not available.
    private double latitude;
    private double longitude;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        Bundle extras = getIntent().getExtras();
        if (extras != null) {
            latitude = extras.getDouble("latitude", 0);
            longitude = extras.getDouble("longitude", 0);
        }
    }

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

    private void setUpMapIfNeeded() {
        if (mMap == null) {
            MapFragment mapFragment = (MapFragment) getFragmentManager().findFragmentById(R.id.map);
            mapFragment.getMapAsync(this);
        }
    }

    @Override
    public void onMapReady(final GoogleMap map) {
        this.mMap = map;
        LatLng latLng = new LatLng(latitude, longitude); // we zoom in to the provided position (or 0, 0) area and put a marker there
        mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 13));
        MarkerOptions mOpt = new MarkerOptions().title(getResources().getString(R.string.geo_position)).position(latLng);
        mMap.addMarker(mOpt);
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.MapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="se.du.hjo.geolocationapi.MapsActivity" />
```

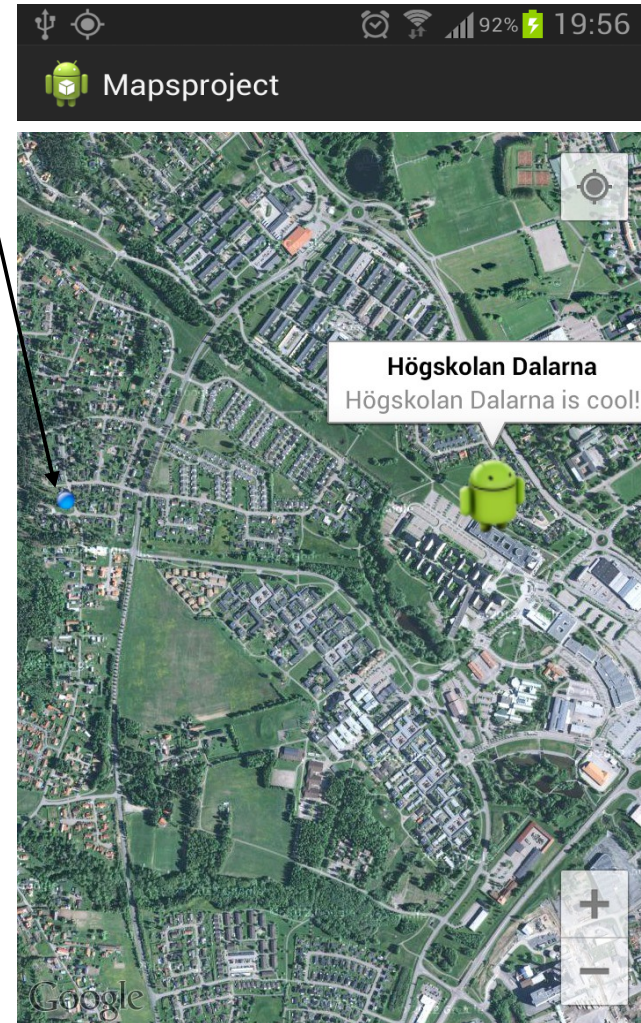
Map view with overlays and Location



- Tap-able Marker and MyLocation (blue point)

```
private GoogleMap mMap;  
private int mZoom = 15;  
private static final LatLng HDA = new LatLng(60.487521, 15.409151);  
// called from onMapReady()  
private void initMapView() {  
    // Enables or disables the my-Location layer and display a nice zoom-in  
    mMap.setMyLocationEnabled(true);  
    // create a marker with information that can be viewed when tapped on  
    Marker hda = mMap.addMarker(new MarkerOptions()  
        .position(HDA)  
        .title("Hogskolan Dalarna")  
        .snippet("Hogskolan Dalarna is cool!")  
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.robot48)));  
    Log.d(TAG, "Checking permissions for Location...");  
    // Check if the Location permission are already available  
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=  
        PackageManager.PERMISSION_GRANTED  
        || ActivityCompat.checkSelfPermission(this,  
        Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {  
        // Location permissions has not been granted  
        requestLocationPermission();  
    }  
    else {  
        Log.d(TAG, "Location permissions has already been granted. Getting location!");  
        Location loc = mLM.getLastKnownLocation(LocationManager.GPS_PROVIDER);  
        if (loc == null)  
            loc = mLM.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);  
        // Move the camera with a zoom of mZoom - 5  
        if (loc != null)  
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(loc.getLatitude(),  
loc.getLongitude()), (float) mZoom - 5));  
        else  
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(HDA, (float) mZoom - 5));  
        // Zoom in, animating the camera for 3 seconds.  
        mMap.animateCamera(CameraUpdateFactory.zoomTo(mZoom), 3000, null);  
    }  
    // Toggle other UI settings  
    // UiSettings uis = mMap.getUiSettings();  
    // uis.setCompassEnabled(true);  
    // uis.setMyLocationButtonEnabled(true);  
}
```

Mapsproject example



Map view using Location



- Make the map follow the user - MyLocation (blue point)

```
// callback function onLocationChanged() is called when new location is obtained
private final LocationListener MyLocationListener = new LocationListener()
{
    @Override
    public void onLocationChanged(Location loc)
    {
        Log.d(TAG, "onLocationChanged");
        if (loc != null)
        {
            try
            {
                DecimalFormat df = new DecimalFormat("##.#####");
                String str = "Lat: " + df.format(loc.getLatitude()) + "\nLng: " + df.format(loc.getLongitude());
                mTV1.setText(str);
                // http://stackoverflow.com/questions/13739990/map-view-following-user-mylocationoverlay-type-functionality-for-
                // android-maps
                LatLngBounds bounds = mMap.getProjection().getVisibleRegion().latLngBounds;

                if(!bounds.contains(new LatLng(loc.getLatitude(), loc.getLongitude())))
                {
                    // Move the camera to the user's location if they are off-screen!
                    mMap.animateCamera(CameraUpdateFactory.newLatLng(new LatLng(loc.getLatitude(),
loc.getLongitude())));
                }
            }
            catch(Exception e)
            {
                Log.e(TAG, "LocationExeption", e);
                mTV1.setText(e.toString());
            }
        }
    }
}
```


Map view in a Fragment



```
// http://stackoverflow.com/questions/19353255/how-to-put-google-maps-v2-on-a-fragment-using-viewpager/
public class MapFragment extends Fragment implements OnMapReadyCallback {
    private MapView mMapView;
    private GoogleMap mGoogleMap;
    private double latitude = 60.0;
    private double longitude = 15.0;
    private View rootView;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        rootView = inflater.inflate(R.layout.fragment_maps, container, false);
        mMapView = (MapView) rootView.findViewById(R.id.mapView);
        // needed to get the map to display immediately
        mMapView.onCreate(savedInstanceState);
        mMapView.onResume();
        try {
            MapsInitializer.initialize(getActivity());
        } catch (Exception e) {
            e.printStackTrace();
        }
        return rootView;
    }

    /** init the fragments screens views */
    @Override
    public void onViewCreated(final View view, Bundle savedInstanceState) {
        if (mGoogleMap == null) {
            mMapView.getMapAsync(this);
        }
    }

    // https://developers.google.com/maps/documentation/android-api/map
    @Override
    public void onMapReady(final GoogleMap map) {
        this.mGoogleMap = map;
        //mGoogleMap.setMyLocationEnabled(true);
        LatLng latLng = new LatLng(latitude, longitude);
        mGoogleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 13));
        MarkerOptions mOpt = new MarkerOptions().title(getResources().getString(R.string.geo_position)).position(latLng);
        mGoogleMap.addMarker(mOpt);
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    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" />
</RelativeLayout>
```

Using the Location Services APIs



`com.google.android.gms.location.***`

- To use the the Google Play Services Location APIs
 - Your AVD and project must include the Google Play Services library (as with Google Maps API V2)
- Key Features
 - Fused location provider intelligently manages the underlying location technology
 - Geofencing APIs setup geographic boundaries around specific locations
 - Activity recognition checks the user's current activity – still, walking, cycling, in-vehicle, ...
- API referens (places, and places-ui)
 - <https://developers.google.com/android/reference/com/google/android/gms/location/package-summary>
- See gms location lessons at
 - <http://developer.android.com/training/location/index.html>

Difference to the `android.location.*` API



- The Google Location Services API, part of Google Play Services, provides a more powerful, high-level framework that automatically handles location providers, user movement, and location accuracy.
- It also handles location update scheduling based on power consumption parameters you provide.
- In most cases, you'll get better battery performance, as well as more appropriate accuracy, by using the Location Services API.
- It makes heavy use of Fragments and callbacks from the Google play services client

Lessons

Retrieving the Current Location

Learn how to retrieve the user's current location.

Receiving Location Updates

Learn how to request and receive periodic location updates.

Displaying a Location Address

Learn how to convert a location's latitude and longitude into an address (reverse geocoding).

Creating and Monitoring Geofences

Learn how to define one or more geographic areas as locations of interest, called geofences, and detect when the user is close to or inside a geofence.

Recognizing the User's Current Activity

Learn how to recognize the user's current activity, such as walking, bicycling, or driving a car, and how to use this information to modify your app's location strategy.

Note: the `AppCompatActivity` Base class is for activities that want to use the support-based Fragment and Loader APIs

Fused Location API

com.google.android.gms.location.FusedLocationProviderApi



- Create instance, connect, get a location and disconnect
 - https://github.com/codepath/android_guides/wiki/Retrieving-Location-with-LocationServices-API

```
// Create an instance of GoogleApiClient in onCreate()
if (mGoogleApiClient == null) {
    mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();
}
```

```
protected void onStart() {
    mGoogleApiClient.connect();
    super.onStart();
}
```

```
protected void onStop() {
    mGoogleApiClient.disconnect();
    super.onStop();
}
```

See the full example
**android-google-maps-
demo.7z** with hotchemi
PermissionsDispatcher

```
public class MainActivity extends AppCompatActivity implements
    ConnectionCallbacks, OnConnectionFailedListener {
    ...
    @Override
    public void onConnected(Bundle connectionHint) {
        mLastLocation =
            LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);

        if (mLastLocation != null) {
            mLatitudeText.setText(String.valueOf(mLastLocation.getLatitude()));
            mLongitudeText.setText(String.valueOf(mLastLocation.getLongitude()));
        }
    }
}
```

Sending objects references between activities or fragments 1



- We need to make our objects use either Serializable or Parcelable
- To pass an object reference to another activity with **Serializable**
- In activity A
 - Make sure the class your objects use implements the serializable interface
 - Create a new intent
 - Create a new bundle
 - Use the putSerializable method to add the object to the bundle with a key
 - Put the bundle in the intent by using the putExtras method
 - Start the activity using the startActivity method
- In activity B
 - Declare an object of the type you want passed
 - Use the getIntent method to get the intent
 - Use the getSerializable method to get the the value using the key
 - Cast the returned value from getSerializable and the desired type
 - To get rid of Type safety warnings use `@SuppressWarnings("unchecked")` on a local variable and then assign it a to member variable

Sending objects references between activities or fragments 2



```
// The class of the object we want to pass needs to implement Serializable
public class GoogleMapsData implements Serializable {
    private static final long serialVersionUID = 0L;
    private double lat;
    private double lng;
    private String text;

    public GoogleMapsData(double lat, double lng, String text) {
        this.lat = lat; this.lng = lng; this.text = text;
    }
    // class getter and setters ...
}
// ArrayList with GoogleMapsData objects
ArrayList<GoogleMapsData> mArrGmapsData = new ArrayList<GoogleMapsData>();
// Add a GoogleMapsData object to the array list
GoogleMapsData gmapsData = new GoogleMapsData(mLocation.getLatitude(),
        mLocation.getLongitude(), mMyText);
mArrGmapsData.add(gmapsData);

// Code in Activity A
Intent mapIntent = new Intent(this, MyMapActivity.class);
// Create a Bundle and put GoogleMaps array data into it
Bundle bundleObject = new Bundle();
bundleObject.putSerializable("gmapsdata", mArrGmapsData);
// Put Bundle in to Intent and call to start Activity
mapIntent.putExtras(bundleObject);
```

Sending objects references between activities or fragments 3



```
public class MyMapActivity extends AppCompatActivity {
    private ArrayList<GoogleMapsData> mArrGmapsData;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        mArrGmapsData = new ArrayList<GoogleMapsData>();

        // if the activity is being resumed...
        if(savedInstanceState != null) {
            @SuppressWarnings("unchecked")
            ArrayList<GoogleMapsData> serializable =
                (ArrayList<GoogleMapsData>) savedInstanceState.getSerializable("gmapsdata");
            mArrGmapsData = serializable;
        }
        else { // we have been called from another activity
            //get the Bundle out of the Intent...
            Bundle extras = getIntent().getExtras();
            if(extras != null) {
                @SuppressWarnings("unchecked")
                ArrayList<GoogleMapsData> serializable =
                    (ArrayList<GoogleMapsData>) extras.getSerializable("gmapsdata");
                mArrGmapsData = serializable;
            }
        }
    }
    @Override
    protected void onSaveInstanceState(Bundle outState) { // app has been paused, save state of our ArrayList
        outState.putSerializable("gmapsdata", mArrGmapsData);
        super.onSaveInstanceState(outState);
    }
    @Override
    protected void onRestoreInstanceState(Bundle savedInstanceState) { // app coming back to life
        super.onRestoreInstanceState(savedInstanceState);
        // restore ...
    }
}
// class cont ...
}
```

Sending objects references between activities or fragments 4



- According to the google engineers **Parcelable** is 10 times faster (you should use Parcelable if many objects is used)
 - The beauty of serializable is that you only need to implement the Serializable interface on a class and its children. It is a marker interface, meaning that there is no method to implement, Java will simply do its best effort to serialize it efficiently.
 - The problem with this approach is that reflection is used and it is a slow process. This mechanism also tends to create a lot of temporary objects and cause quite a bit of garbage collection.
- But Parcelable got at least two times the work to implement
 - One of the reasons for this is that we are being explicit about the serialization process instead of using reflection to infer it. It also stands to reason that the code has been heavily optimized for this purpose.
 - Implementing Parcelable is not free. There is a significant amount of boilerplate code and it makes the classes harder to read and maintain.
- <http://www.developerphil.com/parcelable-vs-serializable/>
- A parcelabler (web and plugin) tool for your classes:
<http://www.parcelabler.com/>

Sending objects references between activities or fragments 5



- Passing code are almost the same as with Serializable, we just change putSerializable to **putParcelable*** and getSerializable to **getParcelable***
bundleObject.putParcelableArrayList("gmapsdata", mArrGmapsData);
ArrayList<GoogleMapsDataParcel> mArrGmapsData = extras.getParcelableArrayList("gmapsdata");
- The following additional Parcable methods needs to be implemented in your class

```
public class GoogleMapsDataParcel implements Parcelable { // The class of the object we want to pass needs to implement Parcelable
    private double lat, lng, String text;

    protected GoogleMapsDataParcel(Parcel in) {
        lat = in.readDouble(); lng = in.readDouble(); text = in.readString();
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeDouble(lat); dest.writeDouble(lng); dest.writeString(text);
    }

    @SuppressWarnings("unused")
    public static final Parcelable.Creator<GoogleMapsDataParcel> CREATOR = new Parcelable.Creator<GoogleMapsDataParcel>() {
        @Override
        public GoogleMapsDataParcel createFromParcel(Parcel in) {
            return new GoogleMapsDataParcel(in);
        }

        @Override
        public GoogleMapsDataParcel[] newArray(int size) {
            return new GoogleMapsDataParcel[size];
        }
    };
}
```