



# Android 201

## Developing an Android App

Reto Meier

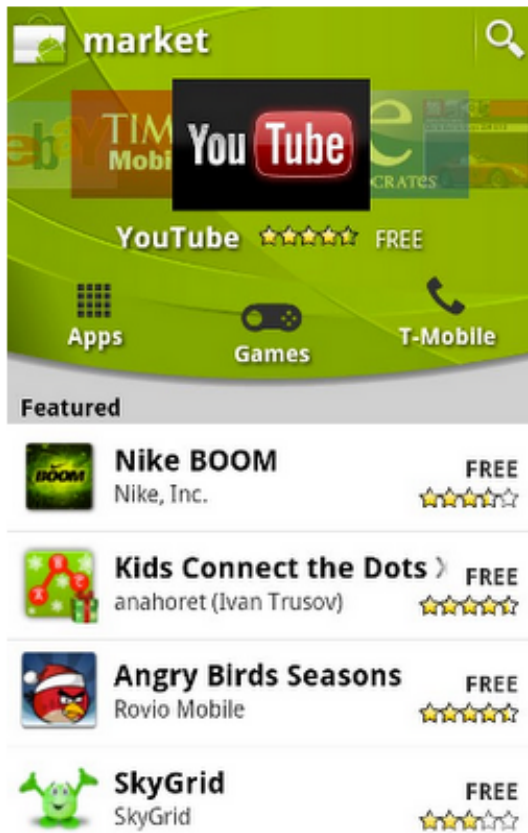
Google  
Android Developer Advocate  
Twitter: @retomeier

# What is Android?



# The Android Market

150k apps, 3 billion downloads



# Android Market Publishing

- **Developers have full control**
- **No review process**
- **App is live in 5 minutes**
- **Updates use same mechanism**
- **Access to download and error reports**



# New in Android Market

- **Web Market**
- **Cloud to Device Messaging**
- **License Verification Service**
- **In App Billing**





# Android Development Features

- **Geo Services**

*Positioning, geocoding, and Maps*

- **Data Sharing**

*Share data and functionality between apps*

- **Background Services**

*Background apps, notifications, and multi-tasking*

- **Home Screen Widgets**

- **Universal Search**

*Surface your app's search results*



# New in Android 2.3

- Gyroscope
- Near Field Communications
- Front facing camera
- Download manager
- Mixable audio effects
- SIP stack



# New in Android 3.0 (Honeycomb)

- Holographic Theme
- Action Bar
- Fragments
- Drag & Drop
- New Animation APIs
- Rich Home Screen Widgets



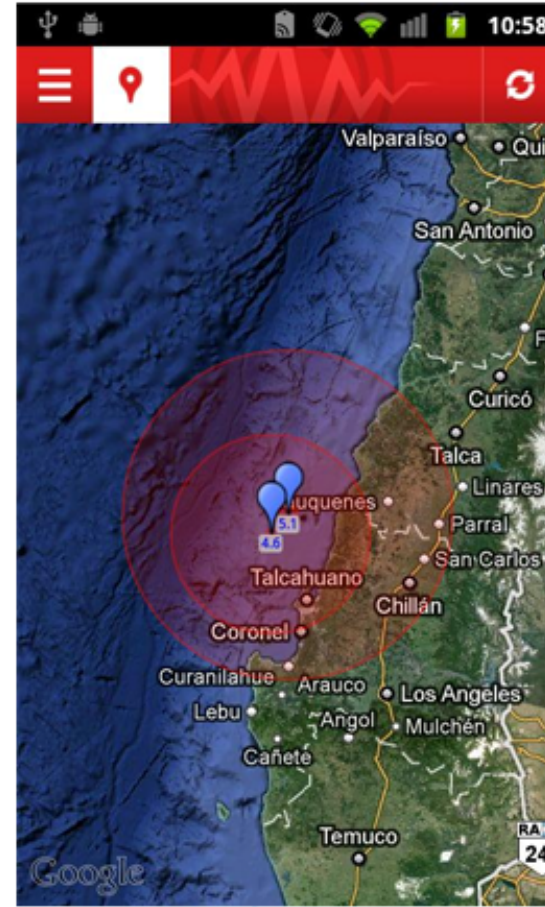
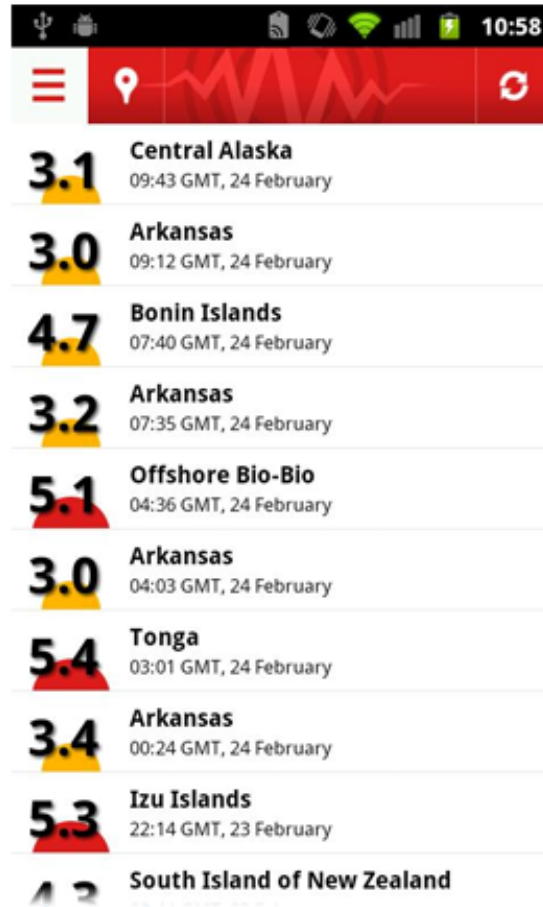


# Building Android Apps

[developer.android.com](https://developer.android.com)



# Phone Apps vs Tablet Apps





# Creating a New Android Application

---



# Hello, Android

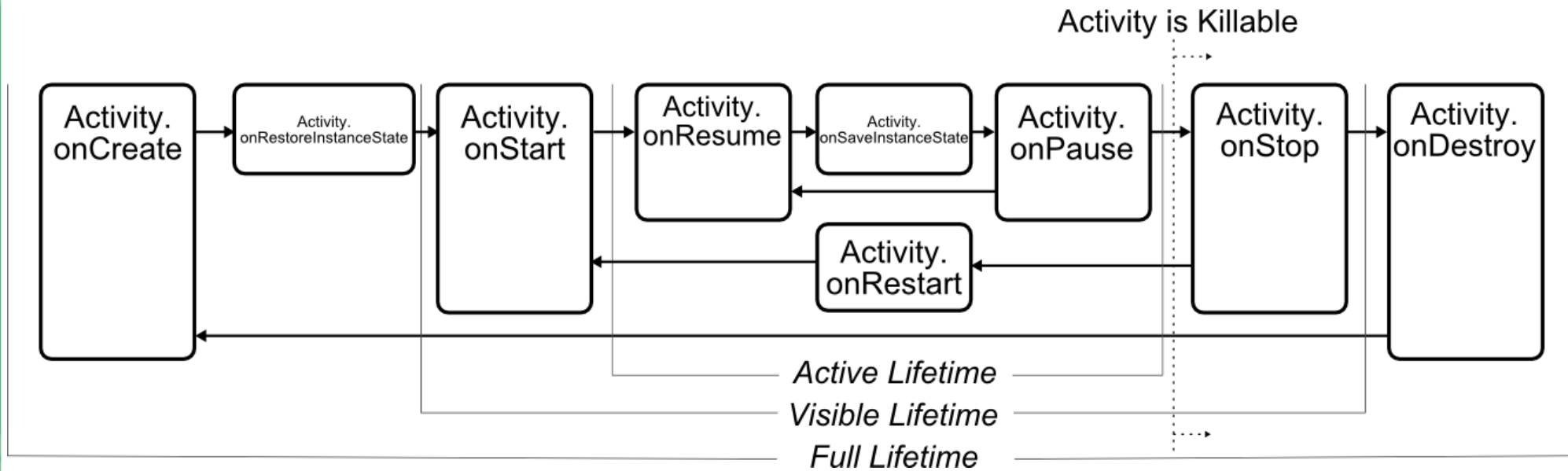
## Activity Class

```
package com.reto.helloandroid;

import android.app.Activity;
import android.os.Bundle;

public class HelloActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

# Android Activity Lifecycle





# Hello, Android

## Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.reto.helloandroid"
  android:versionCode="1"
  android:versionName="1.0">
  <application android:icon="@drawable/icon"
    android:label="@string/app_name">
    <activity android:name=".HelloActivity"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
  <uses-sdk android:minSdkVersion="4" />
</manifest>
```

# Hello, Android

## Layout Resource

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
  />
</LinearLayout>
```

# Hello, Android

## String Resource

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello World, HelloActivity!</string>
  <string name="app_name">Hello Android</string>
</resources>
```



# Earthquake! Getting Started

---



# Earthquake!

## Layout Resource

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <ListView
    android:id="@+id/earthquakeLV"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
  />
</LinearLayout>
```



# Earthquake!

## Parse Quake Feed

```
private void refreshEarthquakes() {
    URL url;
    try {
        String quakeFeed =
            "http://earthquake.usgs.gov/eqcenter/catalogs/1day-M2.5.xml";
        url = new URL(quakeFeed);

        URLConnection connection = url.openConnection();
        HttpURLConnection httpConnection = (HttpURLConnection)connection;
        int responseCode = httpConnection.getResponseCode();

        if (responseCode == HttpURLConnection.HTTP_OK) {
            InputStream in = httpConnection.getInputStream();

            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document dom = db.parse(in);
            Element docEle = dom.getDocumentElement();

            [... Parse the earthquake feed ...]
        }
    }
    [... Exception Handling ...]
}
```

# Earthquake!

## Quake Class

```
public class Quake {  
    private Date date;  
    private String details, link;  
    private Location location;  
    private double magnitude;  
  
    [... public getters ...]  
  
    public Quake(Context context, Date _d, String _det,  
                Location _loc, double _mag, String _link){  
        date = _d;  
        details = _det;  
        location = _loc;  
        magnitude = _mag;  
        link = _link;  
    }  
  
    @Override  
    public String toString() {  
        SimpleDateFormat sdf = new SimpleDateFormat("HH.mm");  
        return sdf.format(date) + ": " + magnitude + " " + details;  
    }  
}
```

# Earthquake!

## Add New Quake

```
ArrayList<Quake> earthquakes = new ArrayList<Quake>();
```

```
private void addNewQuake(Quake _quake) {  
    earthquakes.add(_quake);  
}
```

```
private void refreshEarthquakes() {  
    earthquakes.clear()  
  
    [... parse and add quakes ...]  
}
```

# Earthquake!

## Binding Data to the ListView

```
ArrayAdapter<Quake> aa;
```

```
@Override
```

```
public void onCreate(Bundle icle) {  
    super.onCreate(icle);  
    setContentView(R.layout.main);
```

```
ListView lv = (ListView)findViewById(R.id.earthquakeLV);
```

```
aa = new ArrayAdapter<Quake>(this,  
    android.R.layout.simple_list_item_1,  
    earthquakes);
```

```
lv.setAdapter(aa);
```

```
refreshEarthquakes();
```

```
}
```

```
private void addNewQuake(Quake _quake) {
```

```
    earthquakes.add(_quake);
```

```
aa.notifyDataSetChanged();
```

```
}
```



# Geo APIs: Geocoding, Location, Maps

---



# Earthquake!

## Geocoding a Location

```
public Quake(Context context, Date _d, String _det,
            Location _loc, double _mag, String _link) {
    date = _d;
    details = _det;
    location = _loc;
    magnitude = _mag;
    link = _link;

    Geocoder gc = new Geocoder(context, Locale.getDefault());
    List<Address> address;
    try {
        address = gc.getLocation(_loc.getLatitude(),
                               _loc.getLongitude(), 1);
        if (address.size() > 0) {
            String country = address.get(0).getCountryName();
            if (country != null)
                details = country;
        }
    } catch (IOException e) {Log.d("QUAKE", e.getMessage());}
}
```

# Earthquake!

## Show it on Google Maps

@Override

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    ListView lv = (ListView)findViewById(R.id.earthquakeLV);
    aa = new ArrayAdapter<Quake>(this,
        android.R.layout.simple_list_item_1,
        earthquakes);
    earthquakeListView.setAdapter(aa);

    earthquakeListView.setOnItemClickListener(
        new OnItemClickListener() {
            public void onItemClick(AdapterView<?> av,
                View view, int idx, long arg3) {
                Location selectedLoc = earthquakes.get(idx).getLocation();

                Uri data = Uri.parse("geo:" +
                    selectedLoc.getLatitude() + "," +
                    selectedLoc.getLongitude());
                String action = Intent.ACTION_VIEW;
                Intent viewIntent = new Intent(action, data);
                startActivity(viewIntent);
            }
        });

    refreshEarthquakes();
}
```

# Earthquake!

## Map Activity Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <com.google.android.maps.MapView
    android:id="@+id/map_view"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:enabled="true"
    android:clickable="true"
    android:apiKey="01B92UkuqQGmKvNVz0kRacOa1ZI9cQ-q4vj8j9g"
  />
</LinearLayout>
```

# Earthquake!

## Map Activity Layout

```
public class EarthquakeMap extends MapActivity {  
    @Override  
    public void onCreate(Bundle icle) {  
        super.onCreate(icle);  
        setContentView(R.layout.earthquake_map);  
    }  
  
    @Override  
    protected boolean isRouteDisplayed() {  
        return false;  
    }  
}
```

# Earthquake!

## Adding the MapActivity to the Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.paad.earthquake">
    <application android:icon="@drawable/icon">
        <activity android:name=".Earthquake"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".EarthquakeMap" android:label="Quake Map"/>

        <uses-library android:name="com.google.android.maps"/>
    </application>
    <uses-permission android:name="android.permission.INTERNET"/>
</manifest>
```

# Earthquake!

## View Map Menu Option

```
static final private int MENU_UPDATE = Menu.FIRST;  
static final private int MENU_MAP = Menu.FIRST+1;
```

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    super.onCreateOptionsMenu(menu);  
    menu.add(0, MENU_UPDATE, Menu.NONE, "Refresh");  
    menu.add(0, MENU_MAP, Menu.NONE, "Show Map");  
    return true;  
}
```

```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    super.onOptionsItemSelected(item);  
    switch (item.getItemId()) {  
        case (MENU_UPDATE): {  
            refreshEarthquakes();  
            return true;  
        }  
        case (MENU_MAP): {  
            Intent mapIntent = new Intent(this, EarthquakeMap.class);  
            startActivity(mapIntent);  
            return true;  
        }  
    }  
    return false;  
}
```

# Earthquake!

## Map Overlay

```
public class EarthquakeOverlay extends Overlay {
    ArrayList<GeoPoint> quakeLocations = new ArrayList<GeoPoint>();

    public EarthquakeOverlay(ArrayList<Quake> quakes) {
        super();
        refreshQuakeLocations(quakes);
    }

    private void refreshQuakeLocations(ArrayList<Quake> quakes) {
        quakeLocations.clear();

        for (Quake quake : quakes) {
            Double lat = quake.getLocation().getLatitude()*1E6;
            Double lng = quake.getLocation().getLongitude()*1E6;
            quakeLocations.add(new GeoPoint(lat.intValue(), lng.intValue()));
        }
    }

    @Override
    public void draw(Canvas canvas, MapView mapView, boolean shadow) {
        [... Draw Map Overlay ...]
    }
}
```

# Earthquake!

## Drawing the Map Overlay

```
private static int rad = 5;

@Override
public void draw(Canvas canvas, MapView mapView, boolean shadow) {
    Projection projection = mapView.getProjection();
    Paint paint = new Paint();
    paint.setARGB(250, 255, 0, 0);
    paint.setAntiAlias(true);
    paint.setFakeBoldText(true);

    if (shadow == false) {
        for (GeoPoint point : quakeLocations) {
            Point myPoint = new Point();
            projection.toPixels(point, myPoint);
            RectF oval = new RectF(myPoint.x-rad, myPoint.y-rad,
                                   myPoint.x+rad, myPoint.y+rad);
            canvas.drawOval(oval, paint);
        }
    }
}
```



# Earthquake!

## The Application Class

```
public class EarthquakeApplication extends Application {  
  
    private static EarthquakeApplication instance;  
    public static EarthquakeApplication getInstance() {  
        return instance; }  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        instance = this;  
    }  
  
    private ArrayList<Quake> quakes = new ArrayList<Quake>();  
  
    public void setQuakes(ArrayList<Quake> _quakes) {  
        quakes = _quakes;  
    }  
    public ArrayList<Quake> getQuakes() { return quakes; }  
}
```

# Earthquake!

## Specifying the Application Class in the Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.paad.earthquake">
  <application android:icon="@drawable/icon"
    android:name="EarthquakeApplication">
    <activity android:name=".Earthquake"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>

    <activity android:name=".EarthquakeMap" android:label="Quake Map"/>

    <uses-library android:name="com.google.android.maps"/>
  </application>
  <uses-permission android:name="android.permission.INTERNET"/>
</manifest>
```

# Earthquake!

## Adding the Map Overlay

```
public class EarthquakeMap extends MapActivity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.earthquake_map);  
  
        MapView eqMap = (MapView)findViewById(R.id.map_view);  
        List<Overlay> overlays = earthquakeMap.getOverlays();  
  
        ArrayList<Quake> quakes =  
            EarthquakeApplication.getInstance().getQuakes();  
  
        overlays.add(new EarthquakeOverlay(quakes));  
    }  
  
    @Override  
    protected boolean isRouteDisplayed() {  
        return false;  
    }  
}
```

# Earthquake!

Set new earthquake array

```
private void addNewQuake(Quake _quake) {  
    earthquakes.add(_quake);  
    aa.notifyDataSetChanged();  
  
    EarthquakeApplication.getInstance().setQuakes(earthquakes);  
}
```

# Earthquake!

## Adding the My Location Overlay

```
public class EarthquakeMap extends MapActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.earthquake_map);

        MapView eqMap = (MapView)findViewById(R.id.map_view);
        List<Overlay> overlays = earthquakeMap.getOverlays();
        ArrayList<Quake> quakes =
            EarthquakeApplication.getInstance().getQuakes();

        overlays.add(new EarthquakeOverlay(quakes));
        overlays.add(new MyLocationOverlay(this, eqMap));
    }

    @Override
    protected boolean isRouteDisplayed() {
        return false;
    }
}
```



# Background: Services, Alarms, Notification

---



# Earthquake!

## Creating the Earthquake Service

```
public class EarthquakeService extends Service {  
    ArrayList<Quake> quakes;  
  
    @Override  
    public void onStart(Intent intent, int startId) {  
        refreshEarthquakes();  
        stopSelf();  
    };  
  
    @Override  
    public IBinder onBind(Intent intent) { return null; }  
  
    private void refreshEarthquakes() {  
        [... Clear quakes / parse feed ...]  
    }  
  
    private void addNewQuake(Quake _quake) {  
        quakes.add(_quake);  
        EarthquakeApplication.getInstance().setQuakes(quakes);  
    }  
}
```

# Earthquake!

## Adding the Earthquake Service to the Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.paad.earthquake">
  <application android:icon="@drawable/icon"
    android:name="EarthquakeApplication">
    <activity android:name=".Earthquake"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>

    <activity android:name=".EarthquakeMap" android:label="Quake Map"/>

    <service android:name=".EarthquakeService"/>

    <uses-library android:name="com.google.android.maps"/>
  </application>
  <uses-permission android:name="android.permission.INTERNET"/>
</manifest>
```



# Earthquake!

Broadcasting completed event

```
public class EarthquakeService extends Service {  
    public static final String REFRESH = "REFRESH";  
    public static final String REFRESHED = "REFRESHED";  
  
    @Override  
    public void onStart(Intent intent, int startId) {  
        refreshEarthquakes();  
        sendBroadcast(new Intent(REFRESHED));  
        stopSelf();  
    };  
  
    [... Existing class ...]  
}
```

# Earthquake!

## Refreshing the List from the Service Broadcast

```
@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setContentView(R.layout.main);
    ListView lv = (ListView)findViewById(R.id.earthquakeLV);
    aa = new ArrayAdapter<Quake>(this,
        android.R.layout.simple_list_item_1,
        earthquakes);
    lv.setAdapter(aa);
    refreshEarthquakes();

    registerReceiver(
        new BroadcastReceiver() {
            @Override
            public void onReceive(Context context, Intent intent) {
                updateEarthquakes();
            }
        },
        new IntentFilter(EarthquakeService.REFRESHED));
}

public void updateEarthquakes() {
    earthquakes= EarthquakeApplication.getInstance().getQuakes();
    aa.notifyDataSetChanged();
}
```

# Earthquake!

Broadcasting refresh request

```
public void refreshEarthquakes() {  
    sendBroadcast(new Intent(EarthquakeService.REFRESH));  
};
```

# Earthquake!

## Starting a Service from a Broadcast Receiver

```
public class EarthquakeReceiver extends BroadcastReceiver {  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Intent startIntent = new Intent(context,  
            EarthquakeService.class);  
        context.startService(startIntent);  
    }  
}
```

# Earthquake!

## Registering the Broadcast Receiver in the Manifest

```
<receiver android:name=".EarthquakeReceiver">  
  <intent-filter>  
    <action android:name="REFRESH" />  
  </intent-filter>  
</receiver>
```

# Earthquake!

## Using Alarms to Trigger a Refresh

PendingIntent op;

```
private void setRepeatingRefresh() {  
    String alarm = Context.ALARM_SERVICE;  
    AlarmManager am = (AlarmManager) getSystemService(alarm);  
    Intent intent = new Intent(EarthquakeService.REFRESH);  
    op = PendingIntent.getBroadcast(this, 0, intent, 0);  
  
    int type = AlarmManager.ELAPSED_REALTIME_WAKEUP;  
    long intvl = AlarmManager.INTERVAL_FIFTEEN_MINUTES;  
    long triggerTime = SystemClock.elapsedRealtime() + intvl;  
  
    am.setInexactRepeating(type, triggerTime, interval, op);  
}
```

```
private void cancelRepeatingRefresh() {  
    String alarm = Context.ALARM_SERVICE;  
    AlarmManager am = (AlarmManager) getSystemService(alarm);  
    am.cancel(op);  
}
```

# Questions?

- <http://developer.android.com>
- Stack Overflow tag: **Android**
- **@retomeier, @androiddev**

