

Intermediate Android



Toby Kurien

- Electronic Engineer, Android Freelancer
- 3 years of Android development, 20+ apps, best has 220k+ downloads (BatteryFu)
- Over 16 years of software development



Works at **Freelancer**

Attended **University of Natal, Durban**

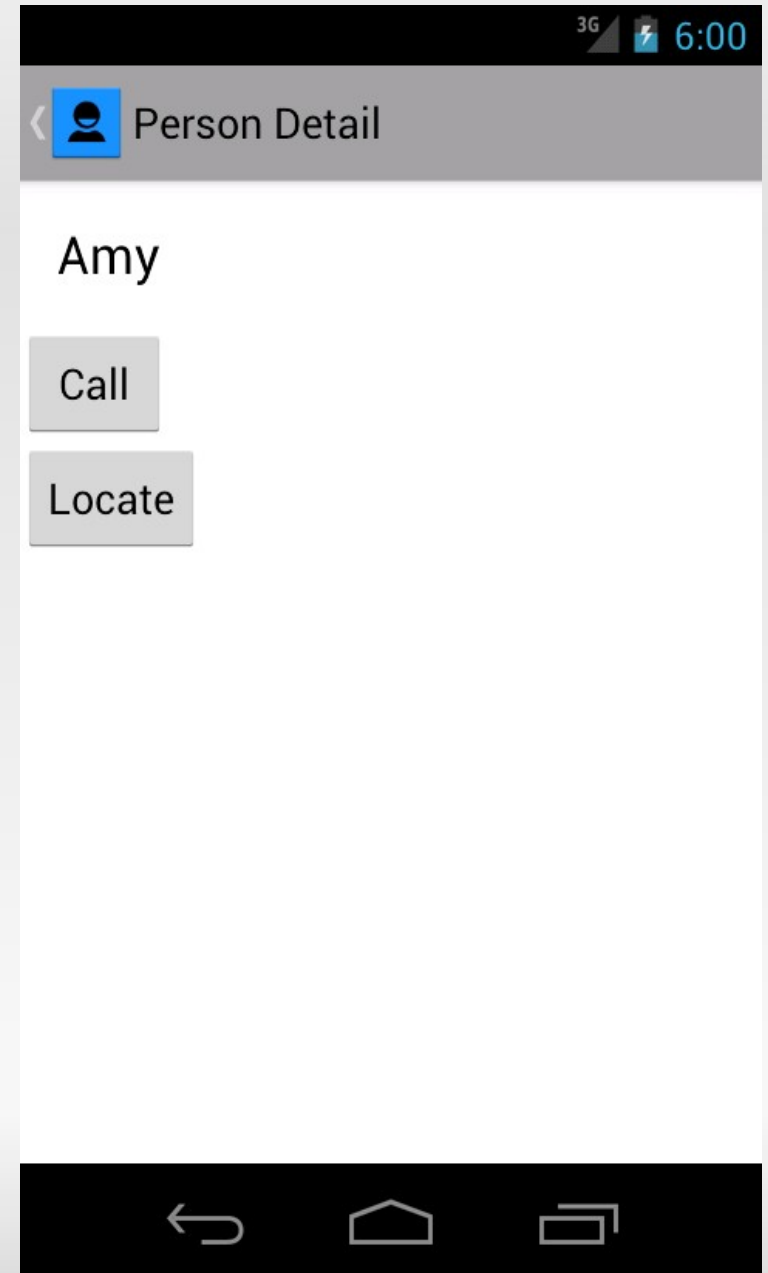
Lives in **Midrand, Johannesburg, South Africa**

Recap of Introductory course

- AndroidManifest.xml
- Activities
- Layouts
- Toasts, LogCat, Exception handling
- Resources: strings, arrays, colors, etc.
- ListView and Adapter
- Network request using HttpURLConnection
- AsyncTask

What we will build

- "MyFamily" app – keep tabs with your family or close friends
- Add contacts as family
- Press button to locate them
- Save details to a database
- Use Android 3+ API – tablet-friendly interface



What you will learn

- Latest Android SDK tools: Strict Mode, Profiler, Lint
- Android 3+ API: Action Bar, Fragments
- ContentProvider
- Sqlite database
- BroadcastReceiver and Notifications
- Intercept/send SMS
- Get GPS location

Environment check

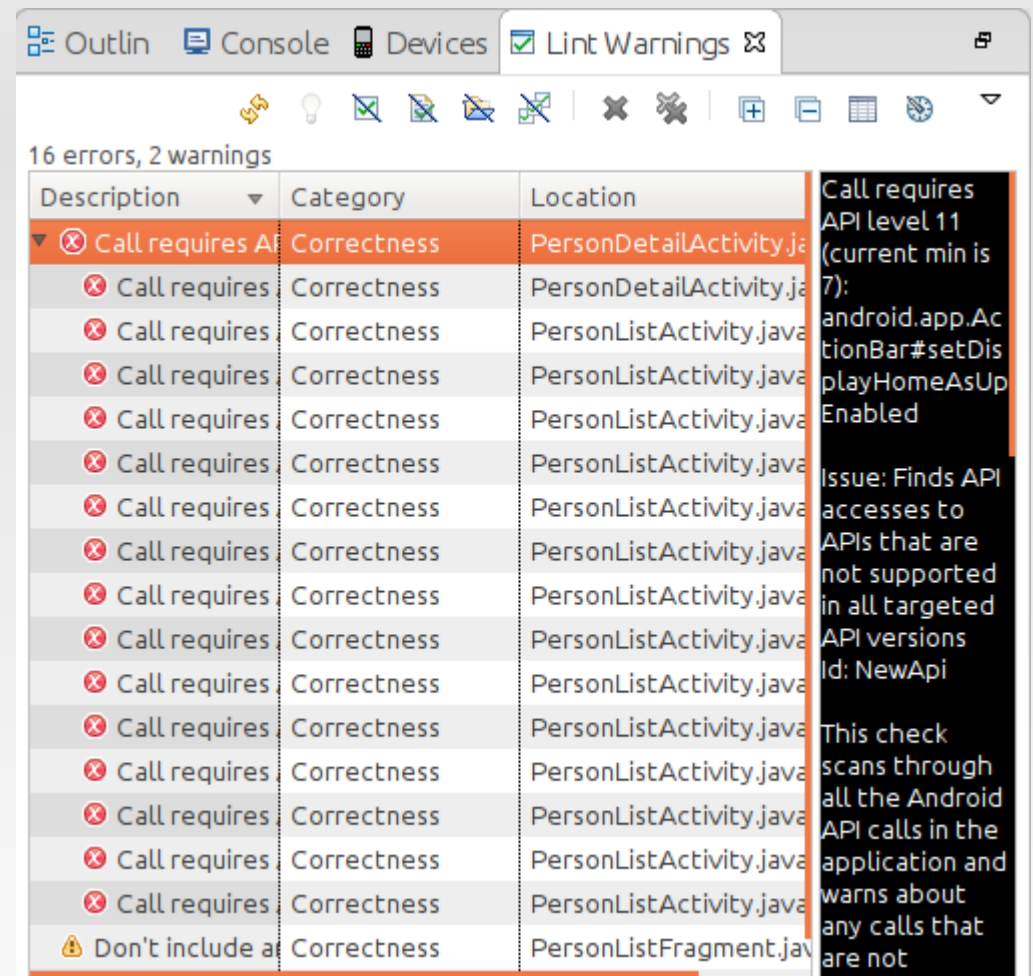
- Eclipse Juno or Kepler
- Android ADT 22
- Android 4.2.2 platform
- Intel image for emulator
 - Significantly faster emulator
 - Use genymotion.com for alternative, fast emulator

Let's get started!

- Create a new Android app
 - App name: MyFamily
 - Package: com.example.myfamily
 - Min SDK: 14, Target SDK: 17
 - Theme: Holo Light with Dark Action Bar
 - Icon: clipart icon
 - Create activity: Master/Detail flow
 - Things: Person, MyFamily

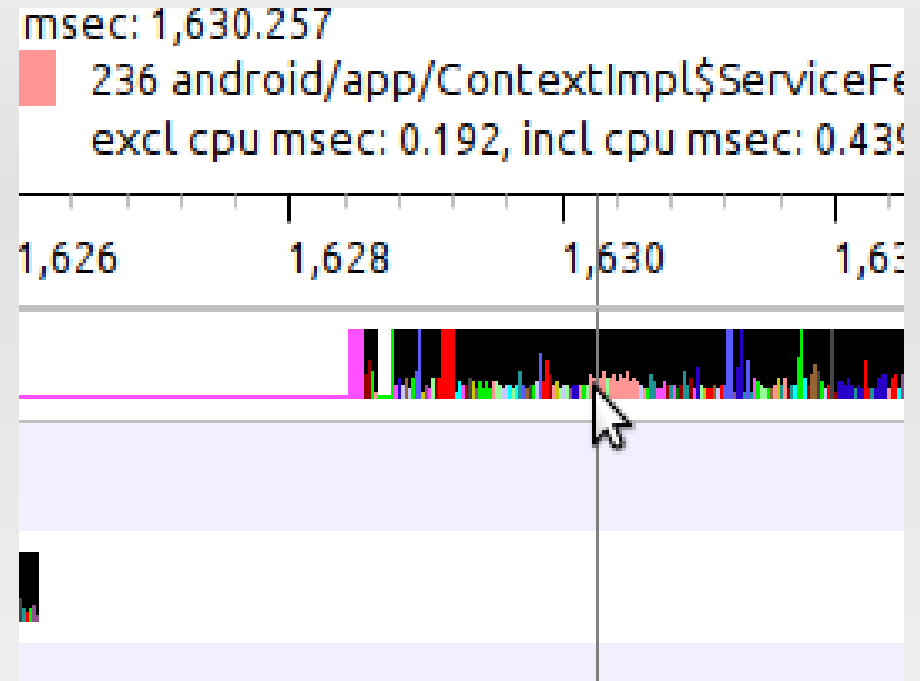
Strict Mode and Lint

- Strict Mode
 - Catches runtime performance and code issues
- Android Lint
 - Catches common Android problems at compile-time
 - Checks for API level issues which is NB!



Profiling

- Method Profiler
 - Shows you how long each method took to run
 - Interactively zoom in and identify slow operations

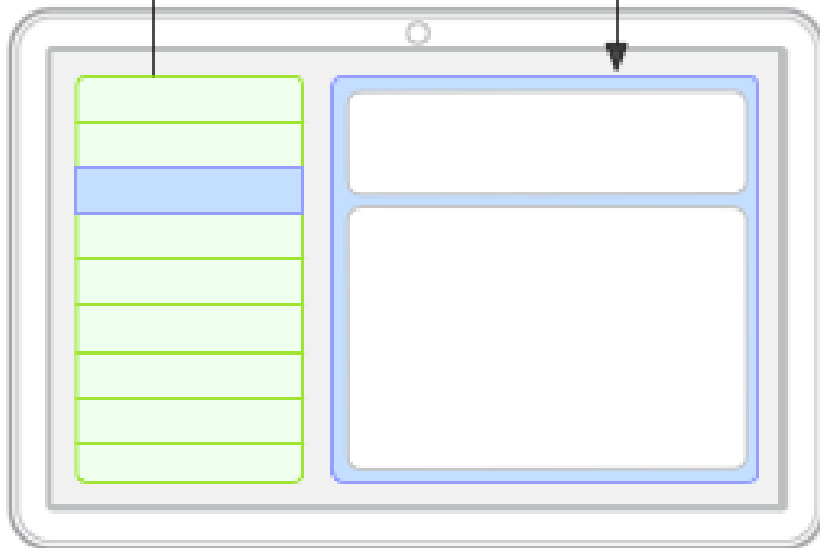


0 (toplevel)	100.0%
1 android/os/Handler.dispatchMessage (Landroid/os/Mes	93.9%
2 android/view/ViewRootImpl.handleMessage (Landroid/v	49.4%
3 android/view/ViewRootImpl.performTraversals ()V	38.0%
4 android/app/ActivityThread\$H.handleMessage (Landro	31.0%
5 android/app/ActivityThread.access\$600 (Landroid/app/	27.4%
6 android/app/ActivityThread.handleLaunchActivity (Lan	27.4%
7 android/app/ActivityThread.performLaunchActivity (La	25.1%

Fragments

Tablet

Selecting an item
updates Fragment B



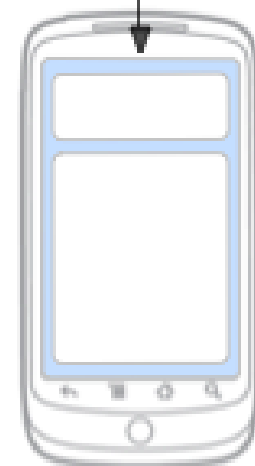
Activity A contains
Fragment A and Fragment B

Handset

Selecting an item
starts Activity B



Activity A contains
Fragment A

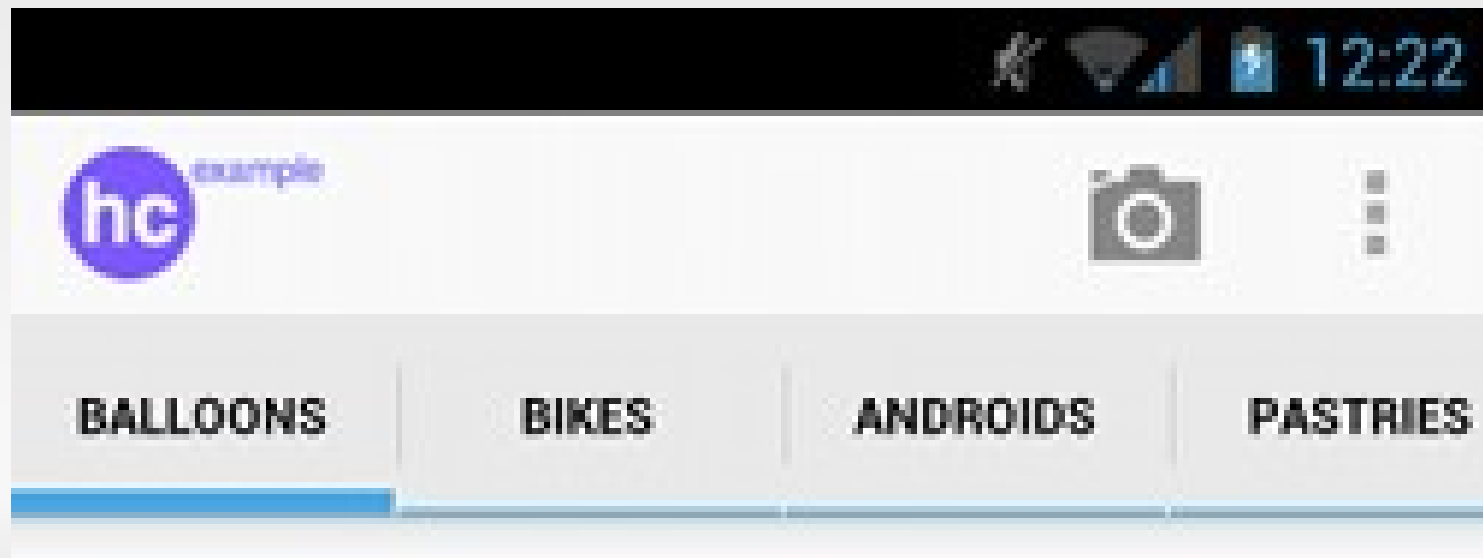
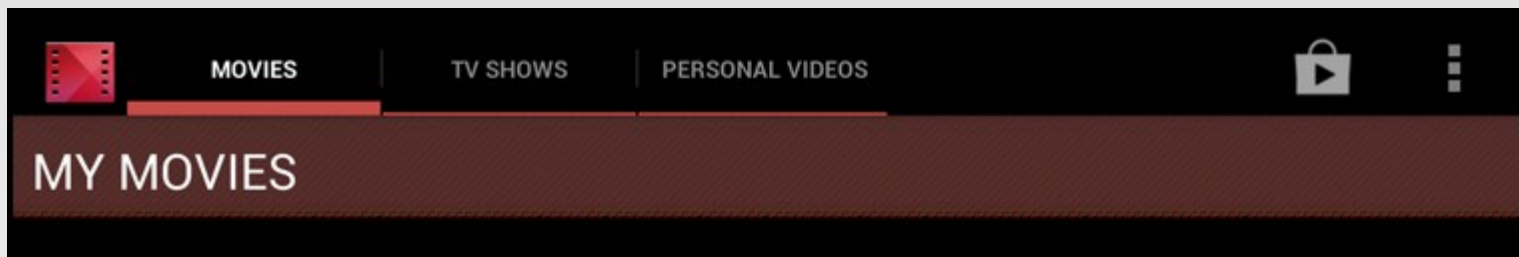
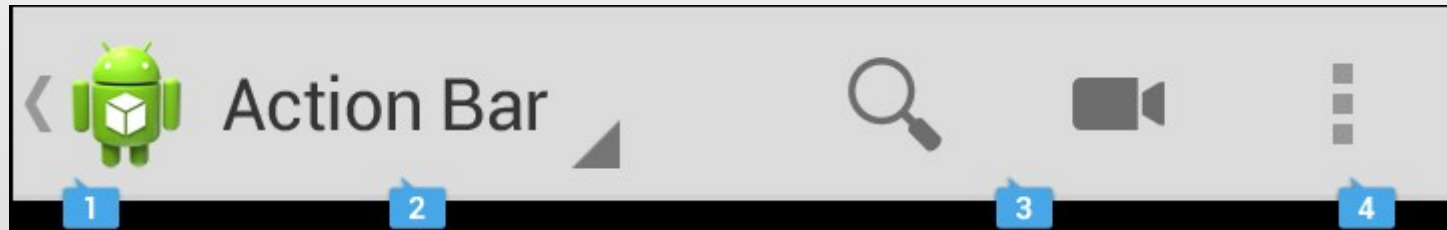


Activity B contains
Fragment B

Fragment lifecycle

- Start up sequence:
 - `onAttach()` → `onCreate()` → `onCreateView()` → `onActivityCreated()` → `onStart()` → `onResume()`
- Shutdown sequence:
 - `OnPause()` → `onStop()` → `onDestroyView()` → `onDestroy()` → `onDetach()`
- From `onDestroyView()` a fragment can re-appear using `onCreateView()`
`onBackPressed`
- `FragmentManager` manages attaching/detaching fragments, back stack

Action Bar



Action Bar resources

- Design guideline:
 - <http://developer.android.com/design/patterns/actionbar.html>
- Developer reference:
 - <http://developer.android.com/guide/topics/ui/actionbar.html>
- Other:
 - <http://actionbarsherlock.com>
 - <http://jgilfelt.github.com/android-actionbarstylegenerator>
 - <http://j.mp/androidassetstudio>

ContentProvider

- Developer guide:
 - <http://developer.android.com/guide/topics/providers/content-provider-basics.html>
- Used to query data sources in Android or in your app
- System content providers:
 - Contacts, Calendar, SMS, MMS
- Android Search uses ContentProvider to search within your own app and other apps

Sqlite database

- Single-file embedded light-weight database with SQL capability
- To work with a database you need:
 - A class to create and version the database tables
→ SQLiteOpenHelper
 - A class to hold the data from the database → Java bean
 - A class to access and transact with the database → DAO class (data access object)
- Android provides helpers and query managers to make it easier

BroadcastReceiver

- Runs when a system or app broadcast Intent is received, e.g. Incoming SMS/call, booted up, battery low, charger connected, etc.
- Will run even if your app is not running
- Can create your own broadcasts
- Terminates as soon as it is complete
- Also see AlarmManager which sends out a broadcast Intent at a specified time (or interval), so you can wake your app up (e.g. Notification of someone's birthday)

Notification

- Status notification with icon
- When clicked, can deep-link into your app via PendingIntent
- Can display a custom view inside notification, e.g. Music player with play/pause buttons



Homework

- Use LocationManager to get a location. Better yet, use the LittleFluffyLibrary to do so.
- Add sending of special location SMS, receiving on the other side in SmsReceiver
- When received, write the person's location to database. This must display on details screen using Google Maps API (or MapIT API)
- Notify user when a location update is received. When clicked, display detail page of that person.

Take it furthur

- Add security - make sure location request is from a person already in your database!
- Unsilence the phone if an SMS or call comes from family. Maybe add continuous notification.
- Implement "locate" menu item to locate all family members. Plot them on a MapView.
- Implement search functionality and add to global search.
- UI: use shape drawables to make background gradients, use 9-patch images to style buttons.

More things you should learn

- Services and IntentServices
 - <http://developer.android.com/guide/components/services.html>
- Use ActionBarSherlock for backwards compatability
 - <http://actionbarsherlock.com/>
- Implementing search functionality and adding to Android global search
 - <http://developer.android.com/guide/topics/search>
- Drawables: Shape, 9-patch (draw9patch), State list:
 - <http://developer.android.com/guide/topics/resources/drawable-resource.html>
- Animation:
 - <http://developer.android.com/guide/topics/resources/animation-resource.html>

Google training resources

- Android design guidelines
 - <http://developer.android.com/design>
- Android training:
 - <http://developer.android.com/training>
- Source code from today's course (incl location and service):
 - <http://github.com/tobykurien/android102>

Thank you!

Toby Kurien

@tobykurien

+Toby Kurien

<http://tobykurien.com>