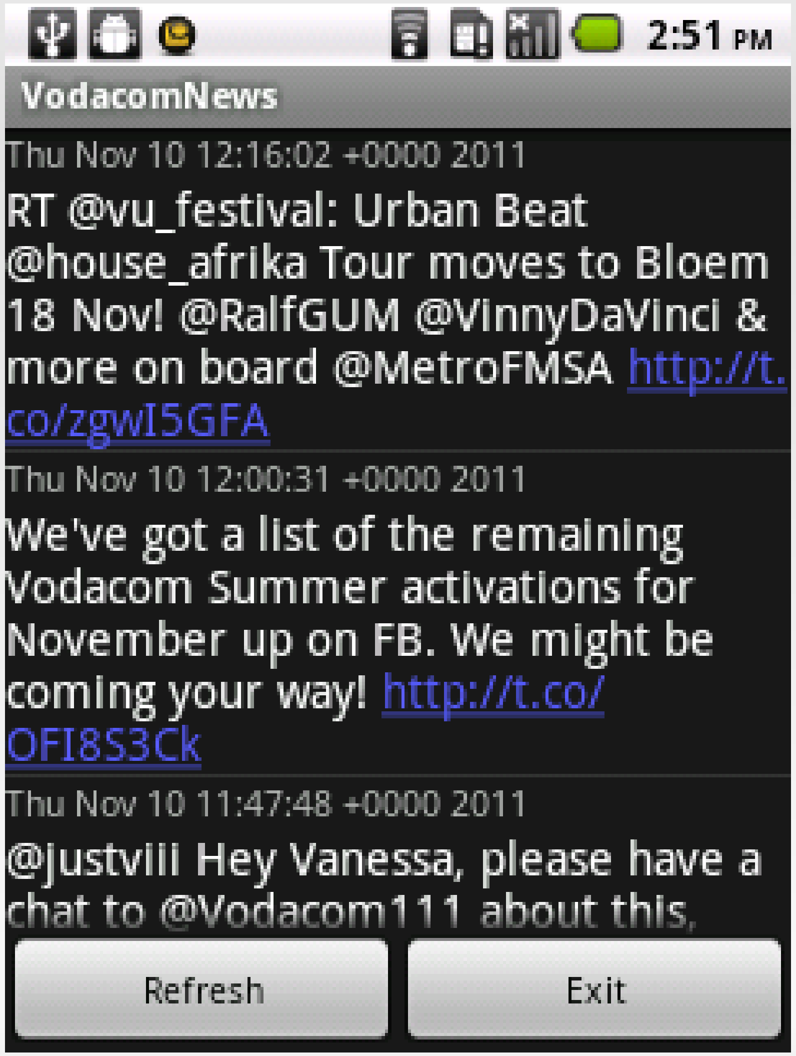
Introduction to Android

Toby Kurien

- Android Freeancer and advocate
- 2 years of Android development, 15+ apps, best has 200k+ downloads (BatteryFu)
- Over 15 years of software development
- Contact me:
  - tobykurien@gmail.com
  - @tobykurien
  - Toby Kurien on Google Plus
  - http://tobykurien.com

What we will build

- Basic twitter client
- Display the timeline of a user
- Multi-threaded web request
- Menu items
- About dialog box
- Settings screen



Thu Nov 10 12:16:02 +0000 2011

RT @vu_festival: Urban Beat @house_afrika Tour moves to Bloem 18 Nov! @RalfGUM @VinnyDaVinci & more on board @MetroFMSA http://t.co/zgwI5GFA

Thu Nov 10 12:00:31 +0000 2011

We've got a list of the remaining Vodacom Summer activations for November up on FB. We might be coming your way! http://t.co/OFI8S3Ck

Thu Nov 10 11:47:48 +0000 2011

@justviii Hey Vanessa, please have a chat to @Vodacom111 about this,

What you will learn

- App icon, drawables
- Android Manifest file
- Activity, UI layout vs code
- Toasts, LogCat, Exception handling
- Layouts: landscape,portrait,etc.
- UI building, LinearLayout, layout weights
- Resources: strings, arrays, colours, dimens

What you will learn

- ListView and Adapter
- Multi-threading, UI Thread, AsyncTask, Handler, Runnable
- HttpUrlConnection and JSON parsing
- Menu items and Intents
- Dialogs
- Settings screen and shared preferences

Let's get started!

- Fire up Eclipse
- Start new Android Project
  - VodacomNews
  - Android 2.3
  - MainActivity
  - za.co.vodacom.android.news

Android Project Folders

- Src – All Java code
- Gen – generated code (the R class)
- Assets – multimedia, data, etc bundled with app
- Bin – compiler output
- Res – all user interface resources
  - Drawable – graphics and drawable xml
  - Layout – all activity layouts
  - Values – strings, colours, styles, etc.
- Android Manifest – app descriptor file

Android Manifest file

- Declares application, activities, services, intents, permissions, hardware required, etc
- Used by Market to filter apps
- Used by installer to create app icon and launch app
- Intent filters tell Android what operations your app can handle (e.g. E-mail, dialer, websites)
- You need to add all activities and persmissions into this file during development

Application icon

- Run the app on your device or emulator
  - Change uses-sdk android:minSdkVersion to 4
  - Enable ”USB Debugging” on device
- App icon is a PNG file in 3 sizes:
  - 72x72 for high density screens (e.g. Tablets)
  - 48x48 for medium density screens (e.g. G1, Hero)
  - 36x36 for low density screens (e.g. Ideos)
- Use an image editor to save 3 versions into the various drawable folders as ic_launcher.png

Connecting the dots

- How does Android know which class and layout to load?
- On Launch, it looks for Intent with action MAIN and category LAUNCHER
- It calls onCreate() method of the class, followed by onStart() and then onResume()
- The rest is up to the class!
- In onCreate() we use setContentView() to load the layout, using the generated R class to reference it as R.layout.main

Button and Toast

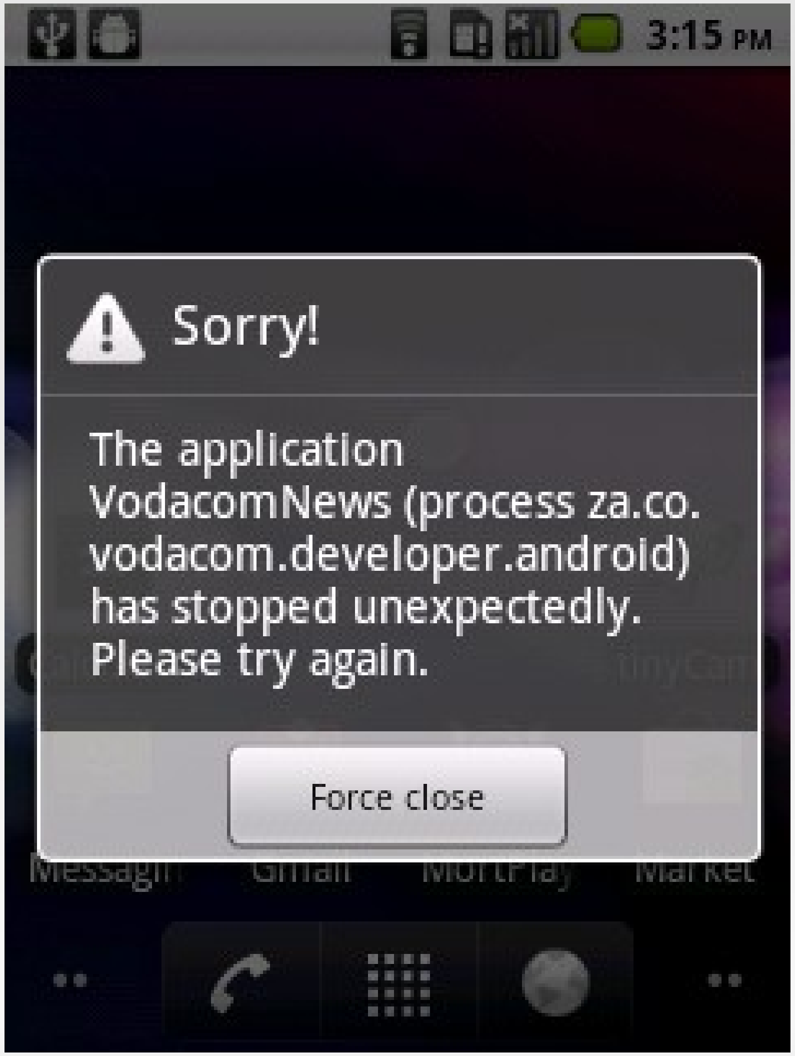
- Add a button to the layout (Button tag)
- Add an onClick handler to the class
- In the onClick, display a Toast

```
[illegible]
  Toast.makeText(this, "Yebo, gogo!",
        Toast.LENGTH_LONG).show();
}
```

- Add a second button

Exceptions and LogCat

- ”Force Close” (FC) vs ”Application Not Responsing” (ANR)
- Create Log tag and use Log.d(...) to output to LogCat
- Create LogCat tab to filter debug messages

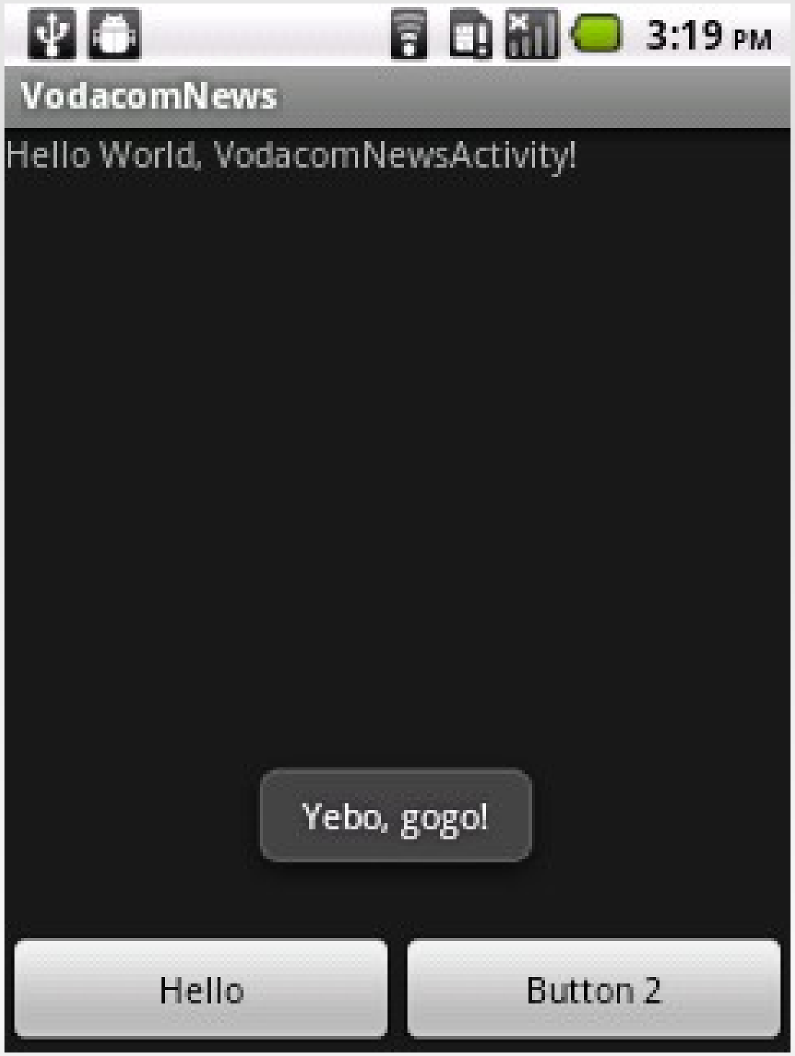


The application VodacomNews (process za.co.vodacom.developer.android) has stopped unexpectedly. Please try again.

Playing with layouts

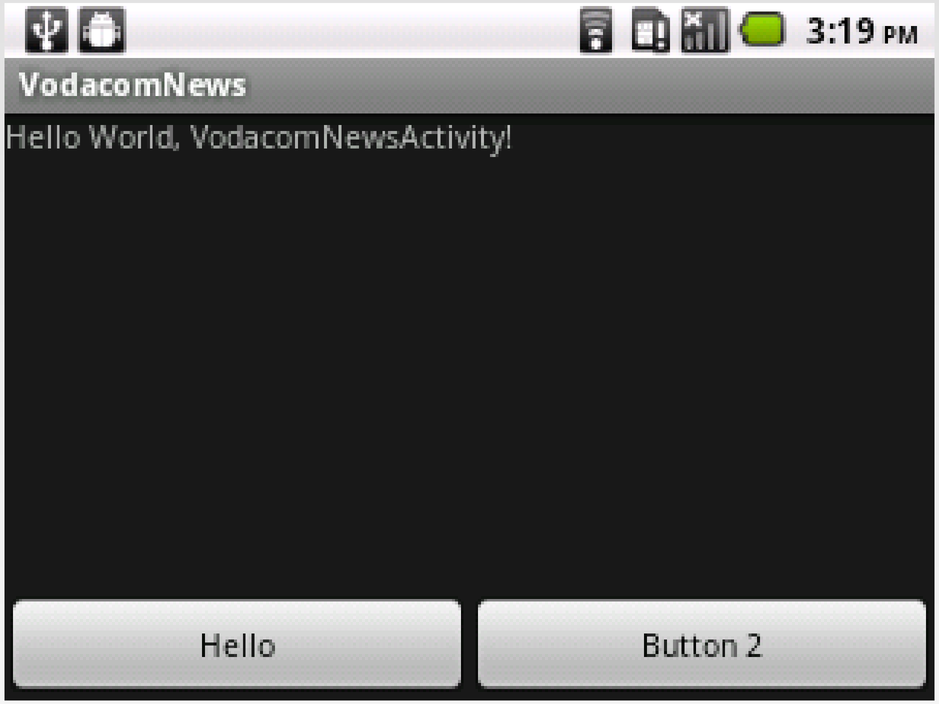
- Align buttons side-by-side with equal width, and at the bottom of the screen (both orientations)
- Use another LinearLayout with horizontal orientation to put the buttons in
- Use layout_weight on the buttons to make them size equally
- Use layout_weight on the TextView, or a ”spring” (dummy item with layout weight) to force buttons to bottom

Playing with layouts


3:19 PM

VodacomNews

Hello World, VodacomNewsActivity!


3:19 PM

VodacomNews

Hello World, VodacomNewsActivity!

Working with Lists

- A ListView is used to display a list of data
- Replace the TextView in main with a ListView
- For entries use @android:array/imProtocols

```
<ListView
  android:id="@+id/main_list"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:entries="@android:array/imProtocols"/>
```

Filling a list with data

- ListView needs 3 things:
  - Layout for each row in the list
  - Data to populate into the layout
  - Adapter to map the data into the layout
- Android provides simple layouts for list rows, e.g. android.R.layout.simple_list_item_1
- Android provides Adapters like SimpleAdapter, ArrayAdapter, CursorAdapter

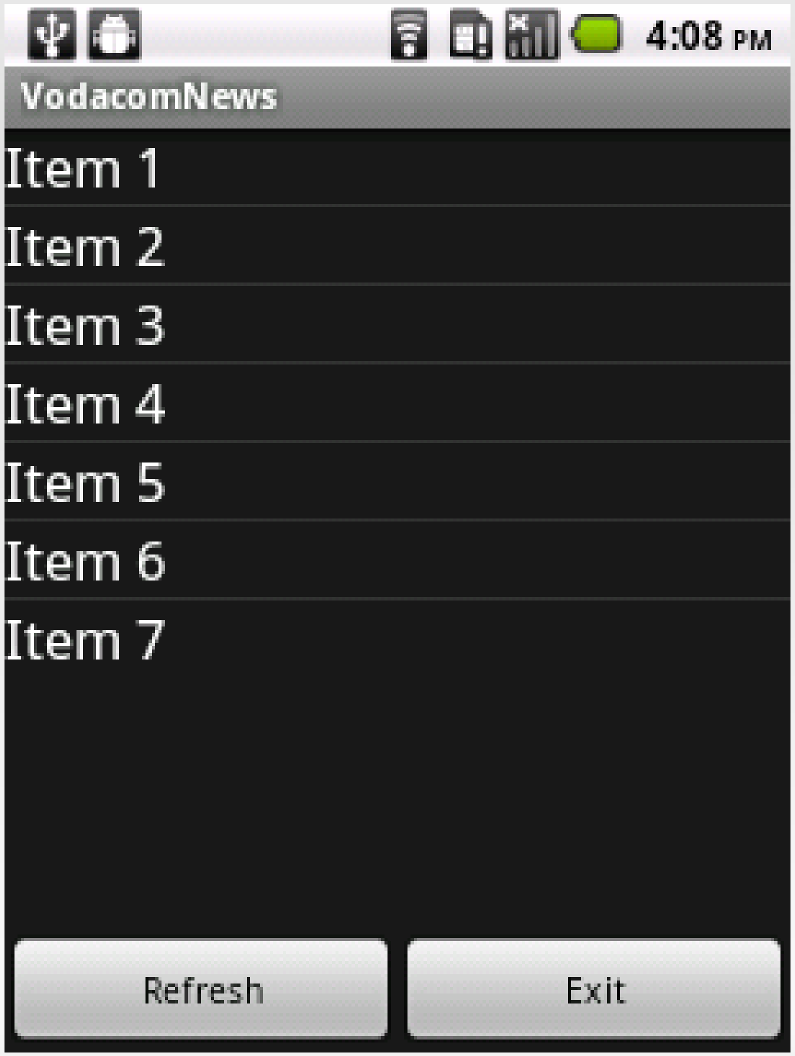
Filling a list with data

- Create layout for list row, main_list_item.xml
  - Simple layout with a TextView item (@+id/main_list_item_text)
- In onCreate() make a String array of data

  ```
  String[] data = new String[]{ ”item 1”, ... };
  ```
- Create an ArrayAdapter<String> passing the context, layout id, TextView id, and data
- Get reference to ListView and set the Adapter

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

Filling a list with data



Custom list Adapter

- Add another TextView in row layout for date/time @+id/main_list_item_date
- Create class Tweet with tweet and date fields
- Create sample tweets in onCreate():

```
Tweet[] data = new Tweet[20];
for (int i=0; i < data.length; i++) {
  data[i] = new Tweet();
  data[i].setTweet(”This is tweet ” + i);
  data[i].setDate(”01/01/2011”);
}
```

Custom list Adapter

- Create NewsAdapter extending BaseAdapter in package .adapter
- Create a constuctor to take context reference and Tweet array of data
- Copy context and data into private members
- Implement getCount() - return size of tweets array
- Implement getItem(int pos) to return data[pos]
- Implement getItemId(int pos) to simply return pos as id of each row

Custom list Adapter

- Implement getView(int pos, View v, ViewGroup parent)
  - If v == null inflate the layout into the view
  - Set the data into the TextViews, and return the view
- Inflating a layout into a View:

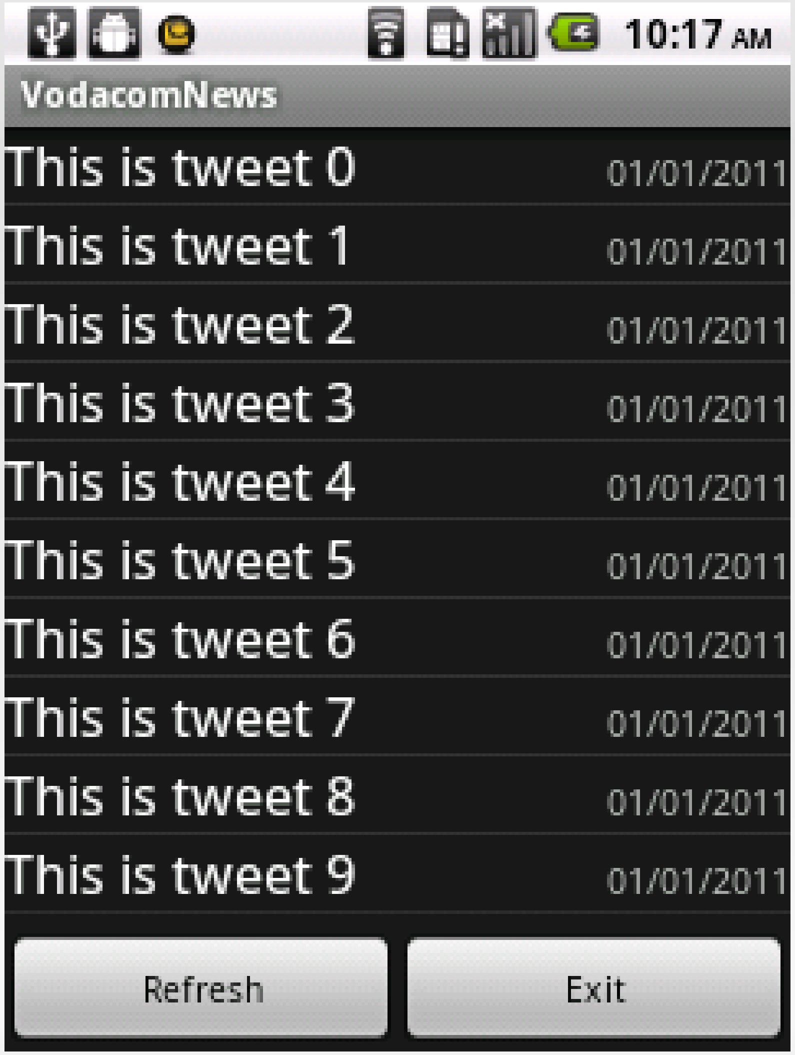
```
v = LayoutInflater.from(context).inflate(R.layout.main_list_item, null);
```

- Setting data into a TextView:

```
Tweet tweet = (Tweet) getItem(pos);
TextView item = (TextView)
   v.findViewById(R.id.main_list_item_text);
item.setText(tweet.getTweet());
```

Custom list Adapter

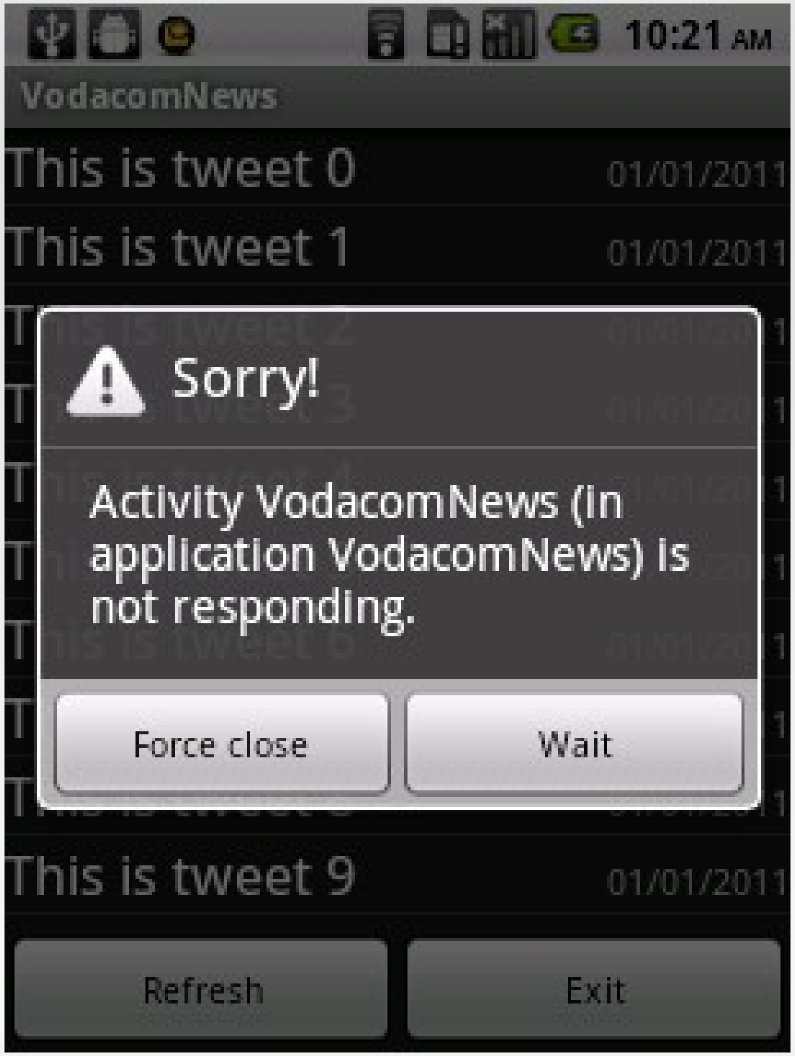
- In onCreate() create the NewsAdapter and set it as the ListView adapter



This is tweet 0 01/01/2011
This is tweet 1 01/01/2011
This is tweet 2 01/01/2011
This is tweet 3 01/01/2011
This is tweet 4 01/01/2011
This is tweet 5 01/01/2011
This is tweet 6 01/01/2011
This is tweet 7 01/01/2011
This is tweet 8 01/01/2011
This is tweet 9 01/01/2011

Multi-threading

- Simulate a slow data loading
  - Add Thread.sleep(1000); into the data creation loop
- Notice the ANR if you press back/menu button
- Notice no UI loads until after onCreate()


10:21 AM
VodacomNews
This is tweet 0 01/01/2011
This is tweet 1 01/01/2011
This is tweet 9 01/01/2011
Sorry!

Activity VodacomNews (in application VodacomNews) is not responding.

Force close
Wait
Refresh
Exit

Multi-threading

- Make new members:

```
Handler handler;
ProgressDialog pd;
```

- Initialise handler = new Handler() in onCreate()
- Show a ”Loading” dialog:

```
pd = new ProgressDialog(this);
pd.setMessage(”Loading...”); // extern this string
pd.show();
```

Multi-threading

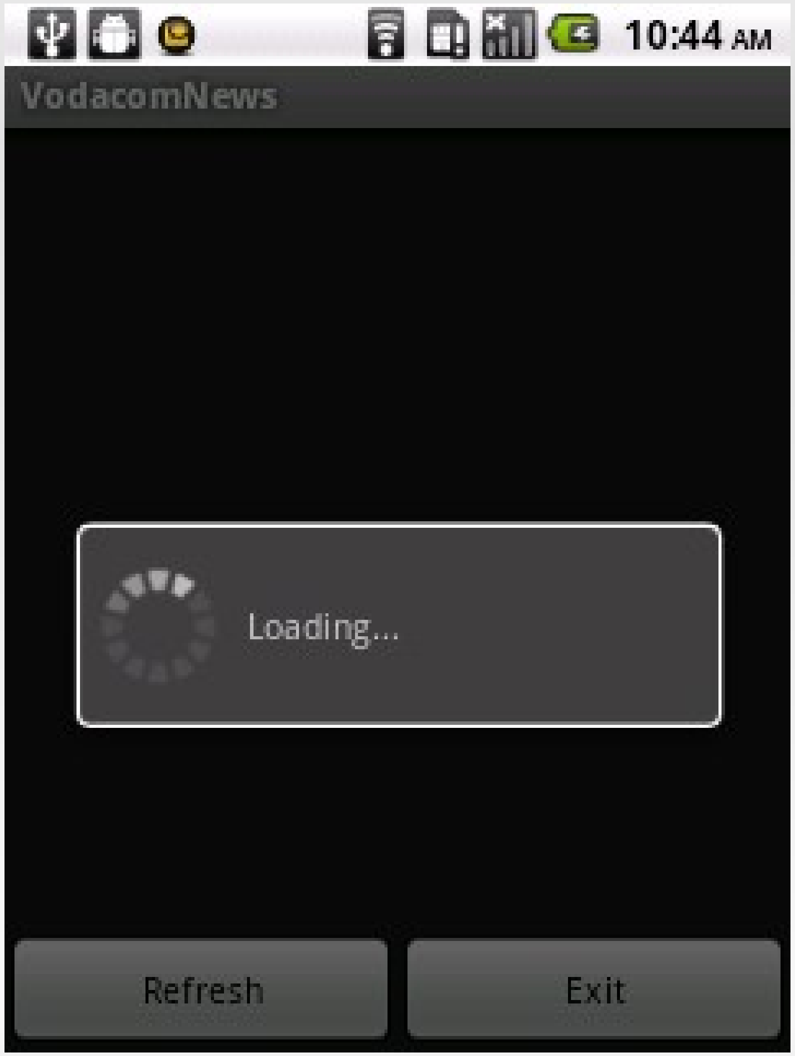
- Create a new AsyncTask. Boilerplate:

```
AsyncTask<Void, Void, Tweet[]> task = new AsyncTask<Void, Void,
Tweet[]>() {
        protected Tweet[] doInBackground(Void... params) {
        }
        protected void onPostExecute(Tweet[] data) {
        }
}
```

- Put data generation into doInBackground()
- Put ListView adapter code into onPostExecute() and add pd.dismiss() once done

Multi-threading

- Voila! Multi-threaded data loading
- Can use handler to update UI from thread
- However – crashes if you rotate during loading!
- Also, reloads data when you rotate!



Handling rotation

- During rotation, Android destroys the activity completely and re-creates it by calling onCreate(), onStart(), onRestart(), and onResume(). Careful with static members!
- In our case, landscape = portait
- Add this to AndroidManifest for this activity:

  android:configChanges="orientation|keyboardHidden"

  - Orientation change is now faster and works!

Parsing JSON data

- Use JSONArray and JSONObject, which work like ArrayList and HashMap

```
JSONArray entries = new JSONArray(jsonData);
data = new Tweet[entries.length()];
for(int i=0; i < entries.length(); i++) {
    JSONObject post = entries.getJSONObject(i);
    data[i] = new Tweet();
    data[i].setTweet(post.getString("text"));
    data[i]setDate(post.getString("created_at"));
}
```

HttpUrlConnection

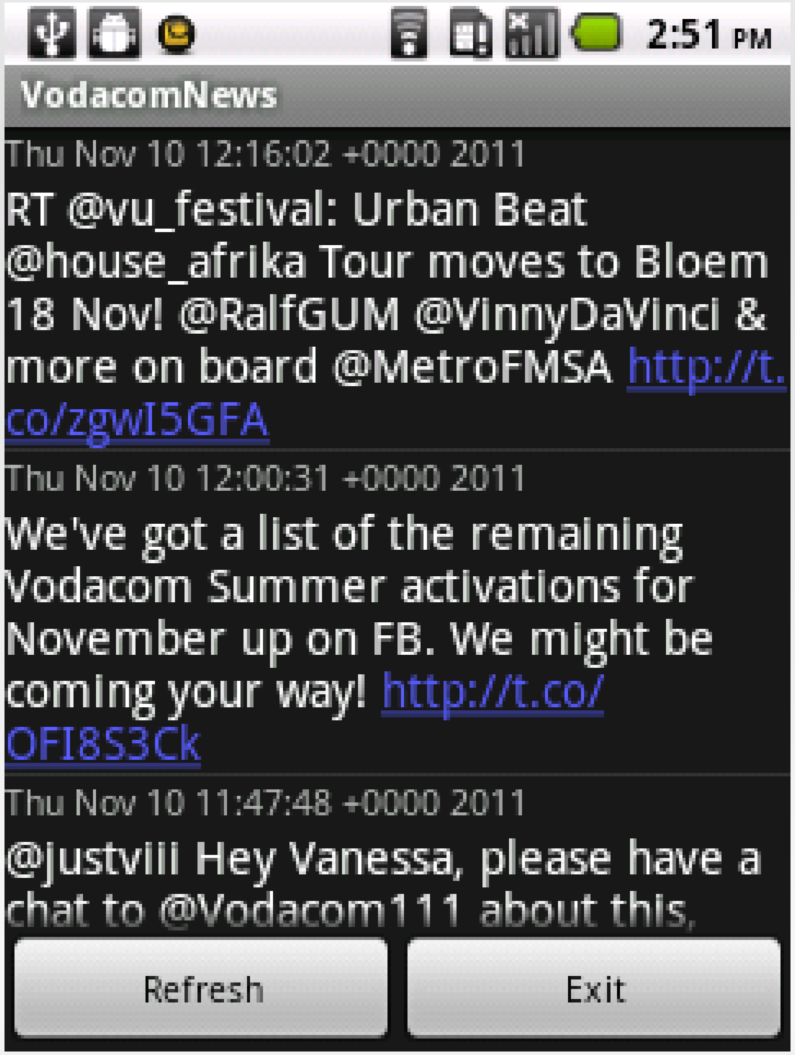
- To get Twitter, we can use HttpUrlConnection
  - URL.openConnection() returns HttpUrlConnection
  - HttpUrlConnection.connect() to connect
  - Check for HttpUrlConnection.getResponseCode() == HttpURLConnection.HTTP_OK
  - HttpUrlConnection.getInputStream() to read data
  - Using the above, implement this:

    public String getData(String url) {…}
  - Use getData to load this URL:

    https://api.twitter.com/1/statuses/user_timeline/vodacom.json

Twitter timeline

- Add INTERNET permission to AndroidManifest.xml
- Once data loaded into data, the rest of the code will display the actual Twitter timeline



Thu Nov 10 12:16:02 +0000 2011
RT @vu_festival: Urban Beat @house_afrika Tour moves to Bloem 18 Nov! @RalfGUM @VinnyDaVinci & more on board @MetroFMSA http://t.co/zgwI5GFA

Thu Nov 10 12:00:31 +0000 2011
We've got a list of the remaining Vodacom Summer activations for November up on FB. We might be coming your way! http://t.co/OFI8S3Ck

Thu Nov 10 11:47:48 +0000 2011
@justviii Hey Vanessa, please have a chat to @Vodacom111 about this,

Adding a Menu

- Create res/menu/main_menu.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
>   <item android:id=”...” android:title=”...” android:icon=”...”/>
</menu>
```

  - Items ”About”, ”Settings”, and ”Exit” using icons: ic_menu_help, ic_menu_preferences, ic_lock_power_off

- Implement onCreateOptionsMenu(Menu menu)

```
MenuInflater inflater = getMenuInflater();
inflater.inflate(R.menu.main_menu, menu);
return super.onCreateOptionsMenu(menu);
```

Adding a menu

- Implement onOptionsItemSelected(MenuItem item)

```
switch (item.getItemId()) {
        case R.id.menu_exit:
           finish();
           return true;
        ...
}
return false;
```

Adding a Dialog

- Create an About dialog for the about menu:

```
public final static int DIALOG_ABOUT = 2;
```

- Implement onCreateDialog(int id) to return a Dialog object when id == DIALOG_ABOUT:

```
Dialog d = new AlertDialog.Builder(this)
    .setTitle("About")
    .setMessage("Hi...")
    .create();
```

- Call showDialog(DIALOG_ABOUT) when menu item is selected

Fun with Intents

- When Settings menu is clicked, try this:

```
Intent i = new Intent();
i.setAction(Intent.ACTION_VIEW);
i.setData(Uri.parse("http://www.vodacom.co.za"));
startActivity(i);
```

- To dial a number:

```
Intent i = new Intent();
i.setAction(Intent.ACTION_DIAL);
i.setData(Uri.parse("tel:1234567"));
startActivity(i);
```

Preference Activity

- Create class SettingsActivity extending PreferenceActivity
- Add activity declaration to Manifest file, with no intent filters
- Call this Intent to start this activity when the Settings menu item is selected:

```
Intent i = new Intent(this, SettingsActivity.class);
startActivity(i);
```

Preference Activity

- Create file res/xml/settings.xml:

```
<PreferenceScreen xmlns:android="
http://schemas.android.com/apk/res/android">
   <ListPreference
     android:key="num_tweets"
      android:title="Max tweets to load"
      android:summary="Maximum number of tweets to load"
      android:entries="@array/max_tweets"
      android:entryValues="@array/max_tweets"
      android:persistent="true"
      android:defaultValue="0"/>
</PreferenceScreen>
```

Shared Preferences

- To use the settings value in your code:

```
SharedPreferences sp = PreferenceManager
  .getDefaultSharedPreferences(this);
int maxTweets = sp.getInt("max_tweets", 0);
```

- Use this by adding this to the Twitter API call URL (if maxTweets > 0): ”?count=” + maxTweets
- To modify the value in code:

```
sp.edit().putInt(”max_tweets”, 10).commit();
```

Spit and polish

- Move onCreate() code into a method to onStart() so that when you go back from SettingsActivity, it reloads the data
- When the ”Refresh” button is clicked, call this method to reload the data
- Remove the second button: make visibility ”gone” in layout
- Done!

Where to from here?

- Read through Android Dev Guide:

  http://developer.android.com/guide/index.html
- Read through Android technical resources:

  http://developer.android.com/resources/index.html
- Read through Toby's corner on AndroidZA:

  http://www.androidza.co.za
- Join the forum on AndroidZA and chat amoung developers!
- Write some free apps and publish them! Best way to learn.

Thanks!

Toby Kurien
tobykurien@gmail.com
@tobykurien
http://tobykurien.com