

# Lesson: From the basics to Android Guru

November 7, 2015

Joel R Sosa



**Disclaimer:** *GDG Puerto Rico is an independent group; our activities and the opinions expressed should in no way be linked to Google, the corporation.*

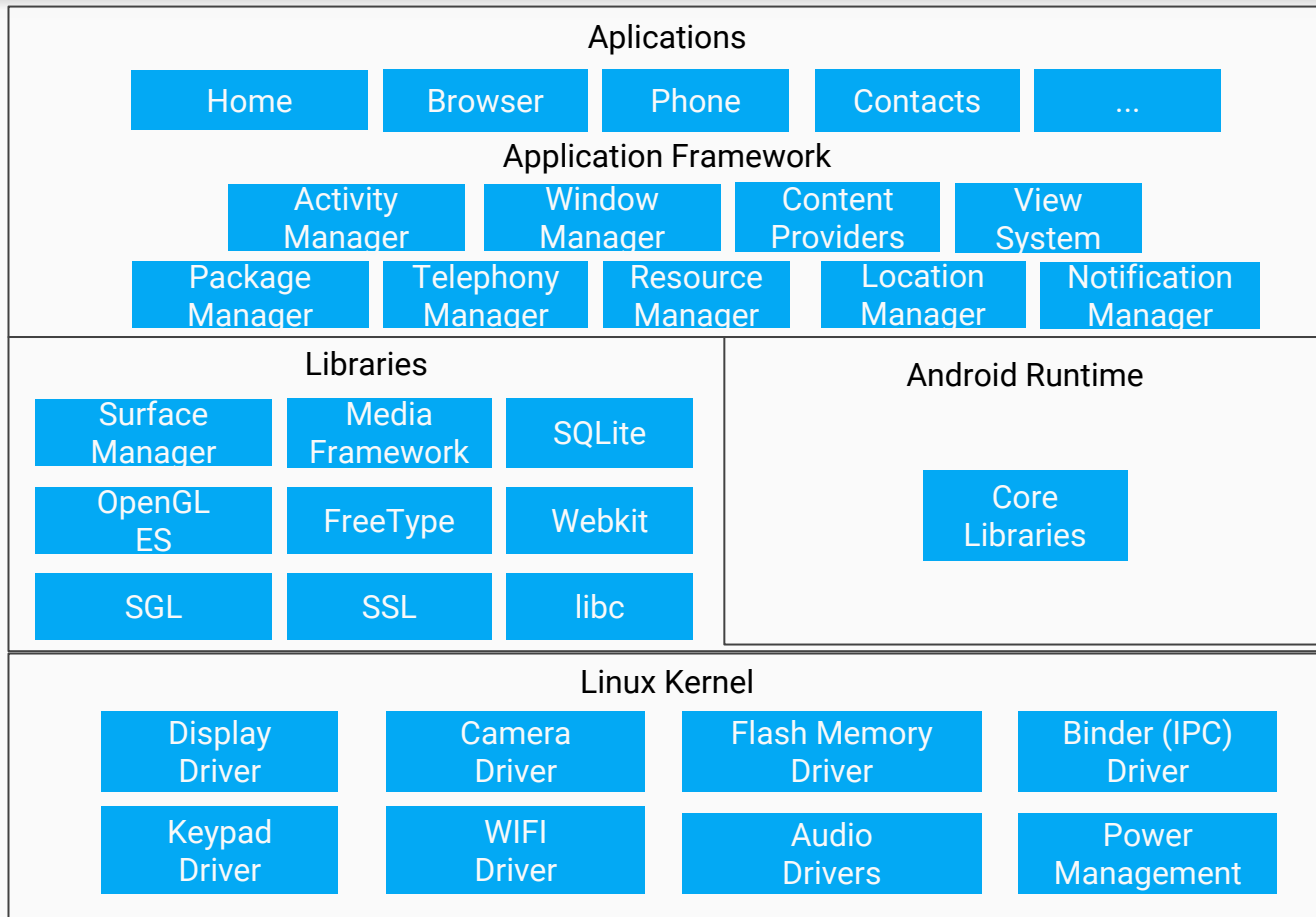
# Agenda

- Where is Android in the world?
- Platform Dissected
- Android Studio
- Basic App
  - Structure aka Directories (layout, resources, values)
  - Types of Resources (strings.xml, colors.xml, styles.xml)
  - Manifest
- More Guru Stuff
  - Intents
  - Adapters

# Where is Android in the world?

- 1 Billion active devices worldwide!
- 1.5 million activated each day!
- 22 version codes and counting!
- “Multi platform” reach
- Written in Java

# Platform Dissection





- Flexible **Gradle**-based build system
- Build variants and *multiple* apk file generation
- Code templates for common features
- Rich layout editor with drag and drop support theme editing
- lint tools to catch performance, version compatibility and more
- ProGuard and app-signing capabilities
- Build-in support for **Google Cloud Platform**

# Basic App, demo time!

Here is what we are going to do!

1. Start a new Android Studio Project
2. Configure your new project
3. Select the form factors our app will run on (phones, wear, tv, etc)
4. Choose a template (Depending your selection here you will add additional info)
5. **Run it!**

Demo

# Intents

- Represents an abstract description of an operation to be performed.
- Used for lots of things
  - Start an **Activity**
  - **Broadcast** an Intent
  - Start a **Service**
  - Communicate with other **apps**
- They can be:
  - *Implicit* - an intent that we don't know which component should be launched. (example: Launching an url, but with what browser?)
  - *Explicit* - an intent that we define its actions, with a specific component. (example: Launching Activity **B** from Activity **A** within our app)



## Example

```
//An implicit intent.  
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.setType("text/plain");  
intent.putExtra(Intent.EXTRA_TEXT, messageText);  
String chooserTitle = getString(R.string.chooser);  
Intent chosenIntent = Intent.createChooser(intent, chooserTitle);  
startActivity(chosenIntent);
```

## Example

```
//An implicit intent.  
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.setType("text/plain");  
intent.putExtra(Intent.EXTRA_TEXT, messageText);  
String chooserTitle = getString(R.string.chooser);  
Intent chosenIntent = Intent.createChooser(intent, chooserTitle);  
startActivity(chosenIntent);
```

## Example

```
//An implicit intent.  
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.setType("text/plain");  
intent.putExtra(Intent.EXTRA_TEXT, messageText);  
String chooserTitle = getString(R.string.chooser);  
Intent chosenIntent = Intent.createChooser(intent, chooserTitle);  
startActivity(chosenIntent);
```

## Example

```
//An implicit intent.  
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.setType("text/plain");  
intent.putExtra(Intent.EXTRA_TEXT, messageText);  
String chooserTitle = getString(R.string.chooser);  
Intent chosenIntent = Intent.createChooser(intent, chooserTitle);  
startActivity(chosenIntent);
```

## Example

```
//An implicit intent.  
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.setType("text/plain");  
intent.putExtra(Intent.EXTRA_TEXT, messageText);  
String chooserTitle = getString(R.string.chooser);  
Intent chosenIntent = Intent.createChooser(intent, chooserTitle);  
startActivity(chosenIntent);
```

## Example

```
//An explicit intent  
Intent intent = new Intent(this, ReceiveIntentActivity.class);  
intent.putExtra(ReceiveIntentActivity.EXTRA_MESSAGE, messageText);  
startActivity(intent);
```

## Example

```
Intent intent = new Intent(this, ReceiveIntentActivity.class);  
intent.putExtra(ReceiveIntentActivity.EXTRA_MESSAGE, messageText);  
startActivity(intent);
```

## Example

```
Intent intent = new Intent(this, ReceiveIntentActivity.class);  
intent.putExtra(ReceiveIntentActivity.EXTRA_MESSAGE, messageText);  
startActivity(intent);
```



## Example

```
Intent intent = new Intent(this, ReceiveIntentActivity.class);  
intent.putExtra(ReceiveIntentActivity.EXTRA_MESSAGE, messageText);  
startActivity(intent);
```

## Example

```
Intent intent = new Intent(this, ReceiveIntentActivity.class);  
intent.putExtra(ReceiveIntentActivity.EXTRA_MESSAGE, messageText);  
startActivity(intent);
```

Demo

# Adapters

- Bridge between UI components
- Used to deliver content.
- Handle Filtering
- Adding new objects to source
- Deleting from source
- Handle different view types

Indirect Subclasses:

`ArrayAdapter<T>`, `BaseAdapter`, `CursorAdapter`, `HeaderViewListAdapter`, `ListAdapter`, `ResourceCursorAdapter`, `SimpleAdapter`, `SimpleCursorAdapter`, `SpinnerAdapter`, `ThemedSpinnerAdapter`, `WrapperListAdapter`

# Example

```
//Android versions represented as an String array, you can use also an ArrayList<T> but it will consume more resources
String[] mVersions = {"Cupcake", "Donut", "Eclair", "Froyo", "Gingerbread", "Honeycomb", "Ice Cream Sandwich",
    "Jelly Bean", "KitKat", "Lollipop", "Marshmallow"};
```

```
//Our adapter binds the information to our View, in this case our ListView.
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, mVersions);
```

```
ListView androidVersionsListView = (ListView) findViewById(R.id.versions_list_view);
androidVersionsListView.setAdapter(adapter);
```

# Example

```
//Android versions represented as an String array, you can use also an ArrayList<T> but it will consume more resources
String[] mVersions = {"Cupcake", "Donut", "Eclair", "Froyo", "Gingerbread", "Honeycomb", "Ice Cream Sandwich",
    "Jelly Bean", "KitKat", "Lollipop", "Marshmallow"};

//Our adapter binds the information to our View, in this case our ListView.
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, mVersions);

ListView androidVersionsListView = (ListView) findViewById(R.id.versions_list_view);
androidVersionsListView.setAdapter(adapter);
```

# Example

```
//Android versions represented as an String array, you can use also an ArrayList<T> but it will consume more resources
String[] mVersions = {"Cupcake", "Donut", "Eclair", "Froyo", "Gingerbread", "Honeycomb", "Ice Cream Sandwich",
    "Jelly Bean", "KitKat", "Lollipop", "Marshmallow"};
```

```
//Our adapter binds the information to our View, in this case our ListView.
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, mVersions);
```

```
ListView androidVersionsListView = (ListView) findViewById(R.id.versions_list_view);
androidVersionsListView.setAdapter(adapter);
```

# Example

```
//Android versions represented as an String array, you can use also an ArrayList<T> but it will consume more resources
String[] mVersions = {"Cupcake", "Donut", "Eclair", "Froyo", "Gingerbread", "Honeycomb", "Ice Cream Sandwich",
    "Jelly Bean", "KitKat", "Lollipop", "Marshmallow"};
```

```
//Our adapter binds the information to our View, in this case our ListView.
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, mVersions);
```

```
ListView androidVersionsListView = (ListView) findViewById(R.id.versions_list_view);
androidVersionsListView.setAdapter(adapter);
```



# Example

```
//Android versions represented as an String array, you can use also an ArrayList<T> but it will consume more resources
String[] mVersions = {"Cupcake", "Donut", "Eclair", "Froyo", "Gingerbread", "Honeycomb", "Ice Cream Sandwich",
    "Jelly Bean", "KitKat", "Lollipop", "Marshmallow"};
```

```
//Our adapter binds the information to our View, in this case our ListView.
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, mVersions);
```

```
ListView androidVersionsListView = (ListView) findViewById(R.id.versions_list_view);
androidVersionsListView.setAdapter(adapter);
```

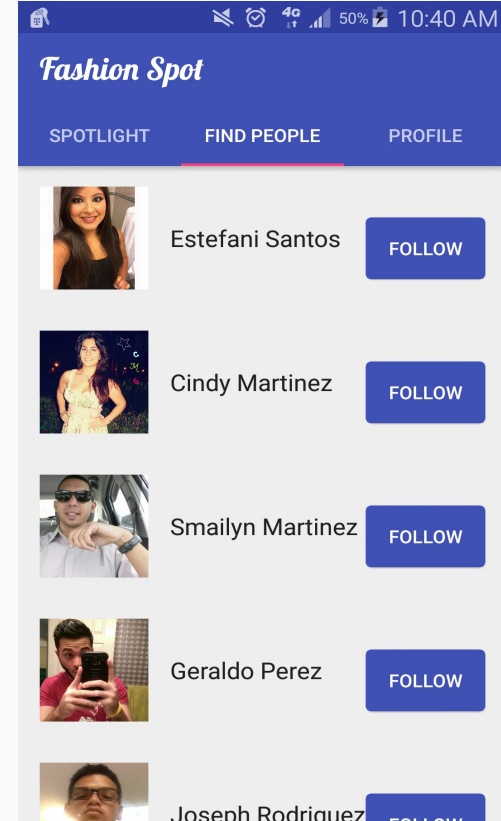
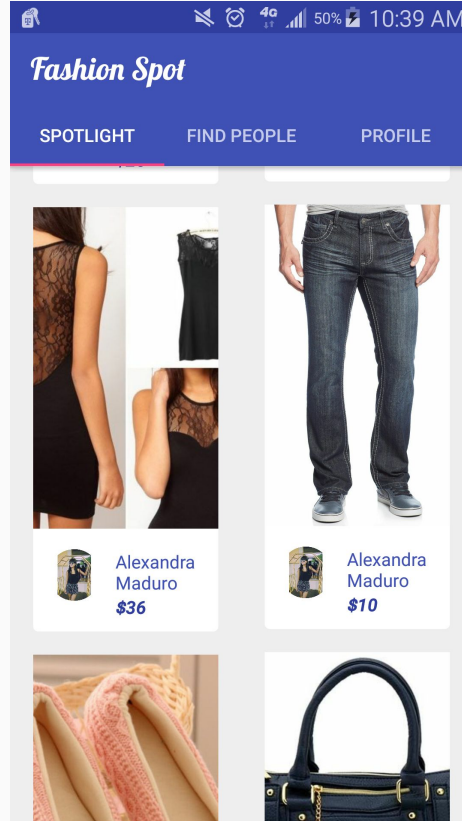
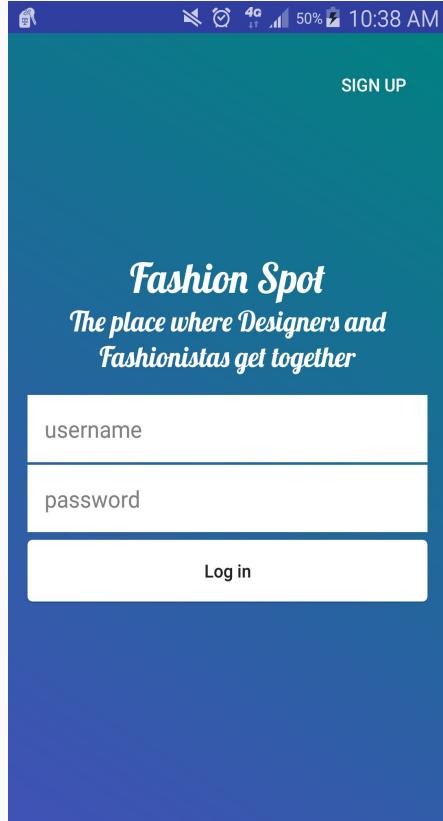
Demo

# There still lot's more!

(too much to present in 30 mins)

- RecyclerView
- CoordinatorLayout
- Snackbars
- <include>'s
- Flavors
- Fragments
- Gradle Tasks
- Unit Testing
- Service
- IntentService
- BroadcastIntent
- Custom Components
- Libraries
- AsynTasks
- Third party API's
- Many many many... more :)

# App examples:



# Time to develop for 1 billion+ users

the journey to guru just started!



@JRSosa



@JoelRSosa