Udacity

Developing Android Apps

Google

Meier, Kuan, Galpin - Summer 2015

Contributors: Max Smith

Latest revision: July 26, 2015

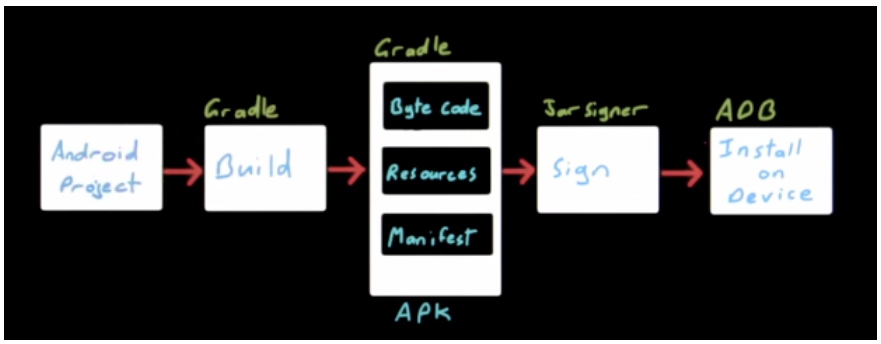
Contents

Abstract

Todo...

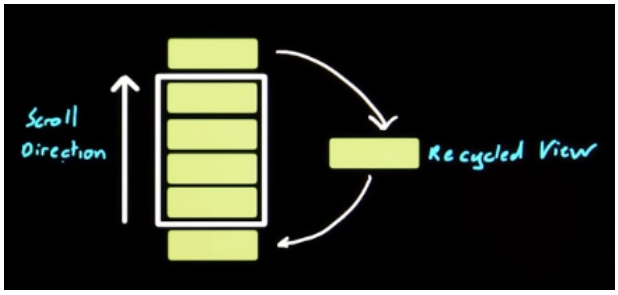
1 Create Project Sunshine

- Package name should be the reverse domain name that you own (eg. com.example - you can't publish a project with this namespae)
- The target SDK is automatically assigned to the most recent version
- MinSDK must should balance a large audience with new features
- An **activity** serves as a presentation layer for our UI
- A **fragment** represents a behaviour or portion of our screen
- To create a new icon:
  1. Right-click→New→Image Asset.
  2. Choose an image
  3. Overwrite "ic_launcher"
- Android Stack:
  - Application Layer
  - Application Framework
  - C/C++ Libs & Android Runtime
  - Linux Kernel
- Gradle is the build toolkit for handling dependencies and building into byte code, resources, and manifest

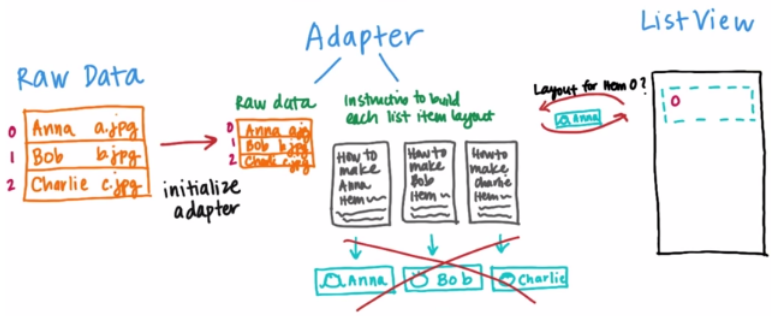


- The "res" file contains all the resource files: layouts, values, images
- Fame, Linear, and Relative layout are basic view groups
- **Frame layout**: useful for simple layouts, with a single view or stack or views. Views are all algined against the frame boundaries only.
- **Linear layout**: perfect for stacking views vertically or horizontally, one after another. Only way to break-up display proportionally.

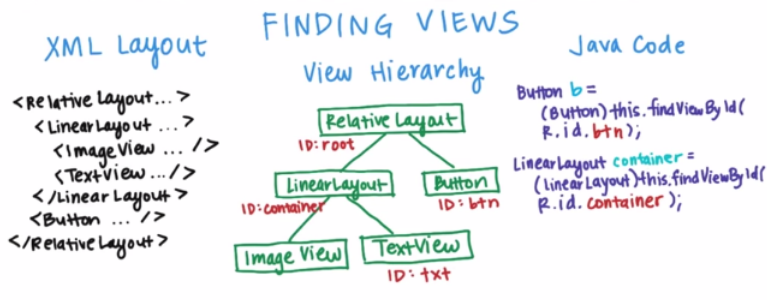
- **Relative layout**: sophisticated layout that allows the position of views relative to other views or the boundaries of the view
- **ScrollView** allows the ability to scroll through the content of the layout it contains
- **ListView** starts by filling the screen with views, and a few on either end to assure no loading on scroll.
- Each view is recycled to save memory when it's moved off screen, which is how all **AdapterView** objects behave



- An adapter then serves an interface for the data, managing location and creation (so memory is dealt with effectively)



- For the sunshine data, which is in array, we'll use an ArrayAdapter, with the paramters:
  - Context: global info about app environment, including access to system servcies: `getActivity`
  - ID of the list item layout: resource id of layout: `R.layout.list_item_forecast`
  - ID of text view: resource id of UI: `R.id.list_item_forecast_textview`
  - List of data: data object `weekForecast`
- ID's aren't required if you don't need a reference to the View/ViewGroup
- View/ViewGroup's are referenced via the `findViewById(ID)` function, with the parameter being the ID of the view in question
- It finds the view by searching through the XML layout's inflated view hierarchy



– You can search through subtrees of the hierarchy by calling the `findViewById` for that particular viewgroup: `container.findViewByID(ID)`

–

```java
// Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
    public void paintComponent(Graphics g) {
        g.drawString("Hello, world!", 65, 95);
    }
}
```

2 Connect Sunshine to the Cloud

3 New Activities and Intents

4 Lifecycle and Databases

5 Content Providers an Loaders

6 Rich and Responsive Layouts

7 Services and Notifications