GETTING GROOVY ON ANDROID

ANDREW REITZ

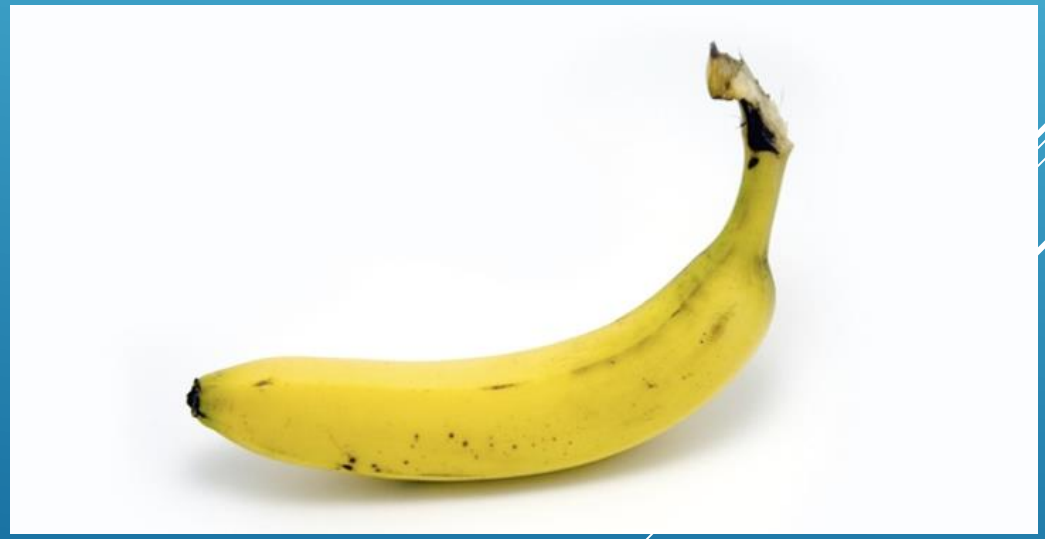
Android Developer @SmartThings

 @AndrewReitz_

 pieces029

OVERVIEW

- Why
- How it works
- How to Android
- Pitfalls
- Other Groovy Awesomeness



WHY ANDROID?

- Fun
  - Different problems
- Free and Open Source
  - I can use linux!
  - $25 to publish an app(s)

WHY GROOVY?

- Eases the Pain of Android Development
  - Less Verbose
  - DSLs
  - AST Transformations
  - Extension Methods

```
public final class Person {
  private final String firstName;
  private final String lastName;
  public Person(String firstName, String lastName) {
    this.firstName = firstName;
    this.lastName = lastName;
  }
  public String getFirstName() { return firstName; }
  public String getLastName() { return lastName; }
  @Override public boolean equals(Object o) { /*lots of code here*/ }
  @Override public int hashCode() { /*lots of code here*/ }
  @Override public String toString() { /*lots of code here*/ }
}
```

WHY GROOVY

```
@Immutable
final class Person {
  private final String firstName
  private final String lastName
}
```

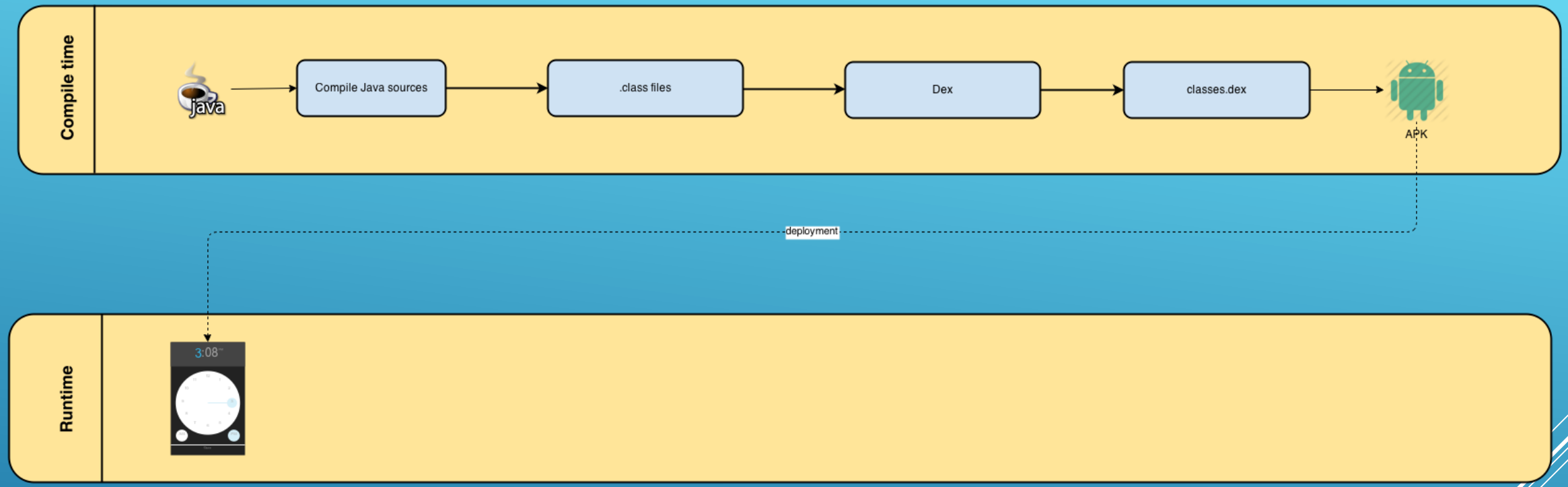
WHY GROOVY

```
button.setOnClickListener(new View.OnClickListener() {
  @Override void onClick(View v) {
    //...
  }
});
```

WHY GROOVY

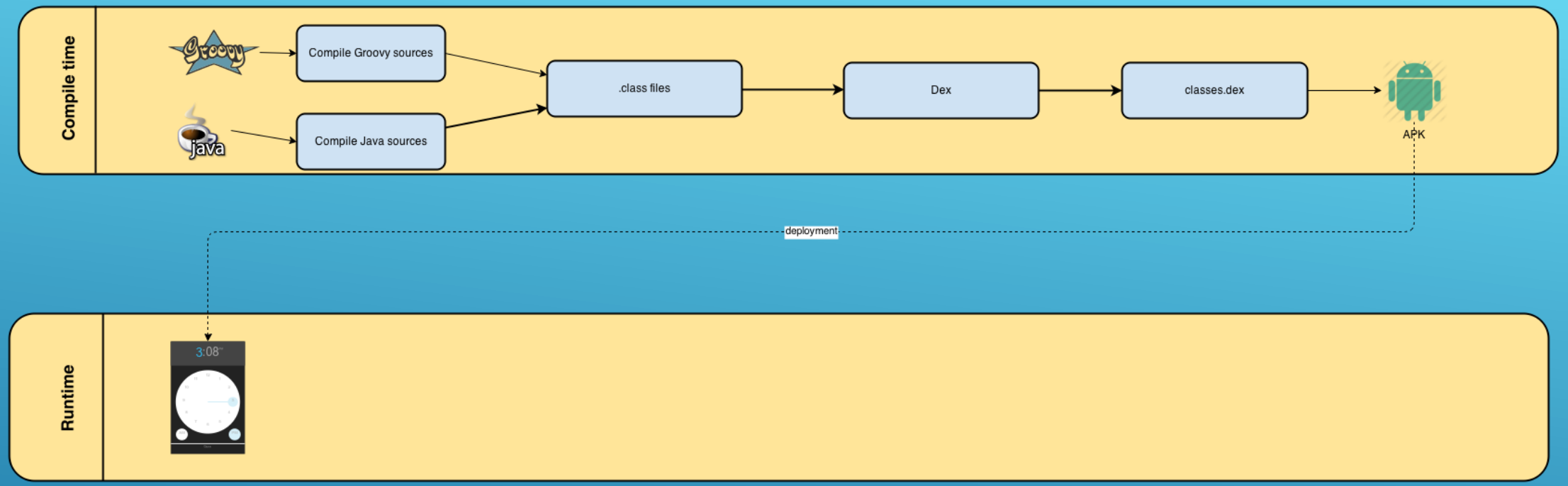
```
button.onClickListener = {
   //...
}
```

WHY GROOVY



HOW IT WORKS

Picture Courtesy of
Cédric Champeau



HOW IT WORKS

Picture Courtesy of
Cédric Champeau

HOW IT WORKS - GOOID JAR

- Android Missing java.bean package
- Workarounds for Android behavior
- Reduced method count

```
buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.1.0'
        classpath 'org.codehaus.groovy:gradle-groovy-android-plugin:0.3.6'
    }
}

apply plugin: 'groovyx.grooid.groovy-android'
```

- Place code in src/main/groovy
- Issue: Android Gradle Plugin Changes

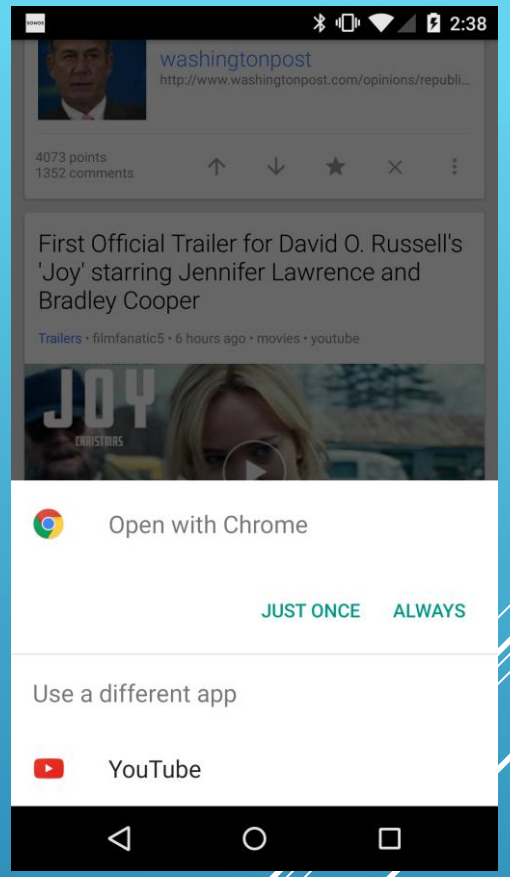
HOW IT WORKS - GRADLE PLUGIN

- Android Manifest
- Application
- Activities
- Fragments
- Views
- Resources

HOW TO ANDROID

HOW TO ANDROID - ANDROIDMANIFEST

- "The manifest file presents essential information about your app to the Android System..."
- Every app must have one
- Tells the system what Activity to start.
- Defines permissions your app needs.
- Tells the system what services you have.
- Tells the system and other apps, what you're app is capable of.
- Example

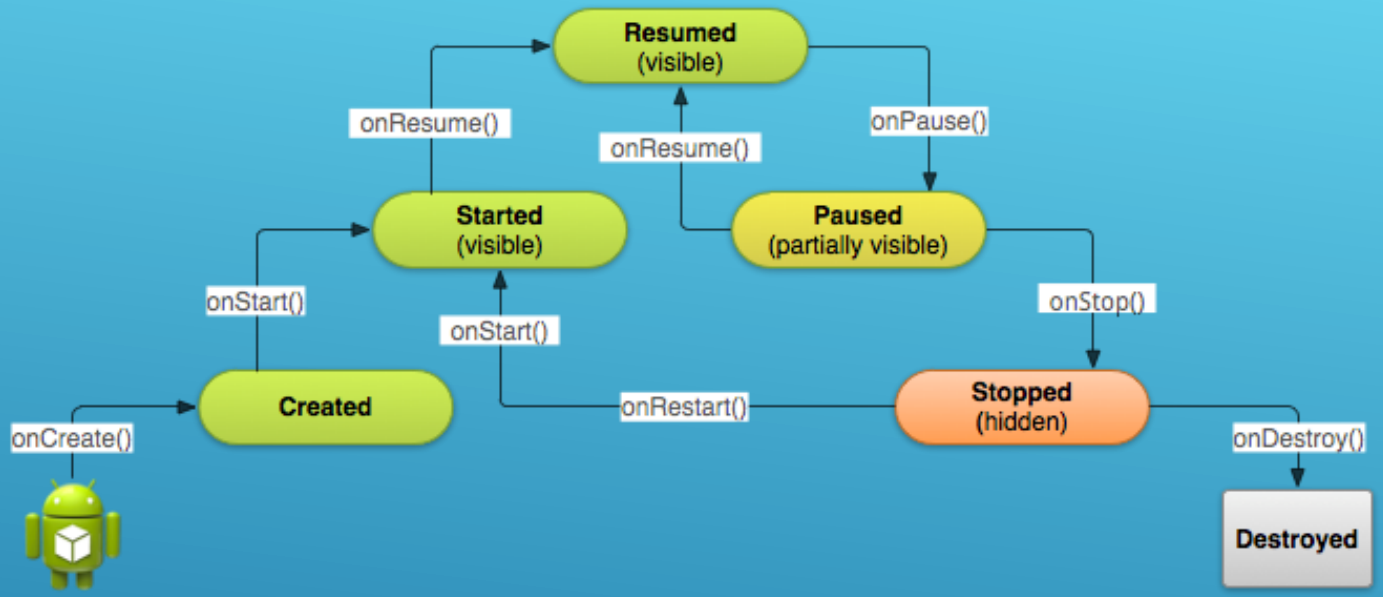


APPLICATION

- Only one
- Also a "Context" (global information about an application environment)
- Bootstrap stuff in onCreate
- Memory management (onTrimMemory, onLowMemory)
- Example

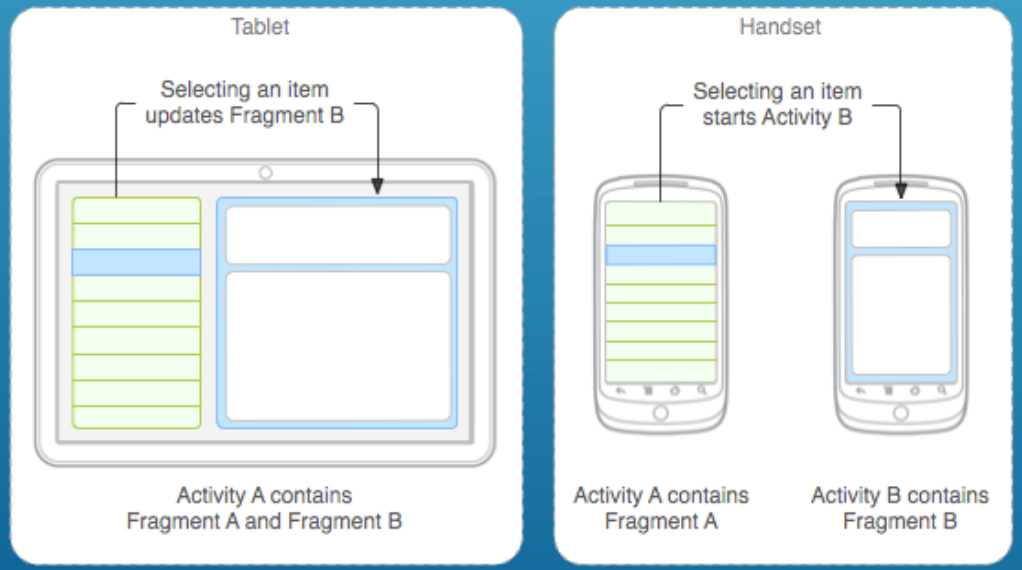
ACTVITIES

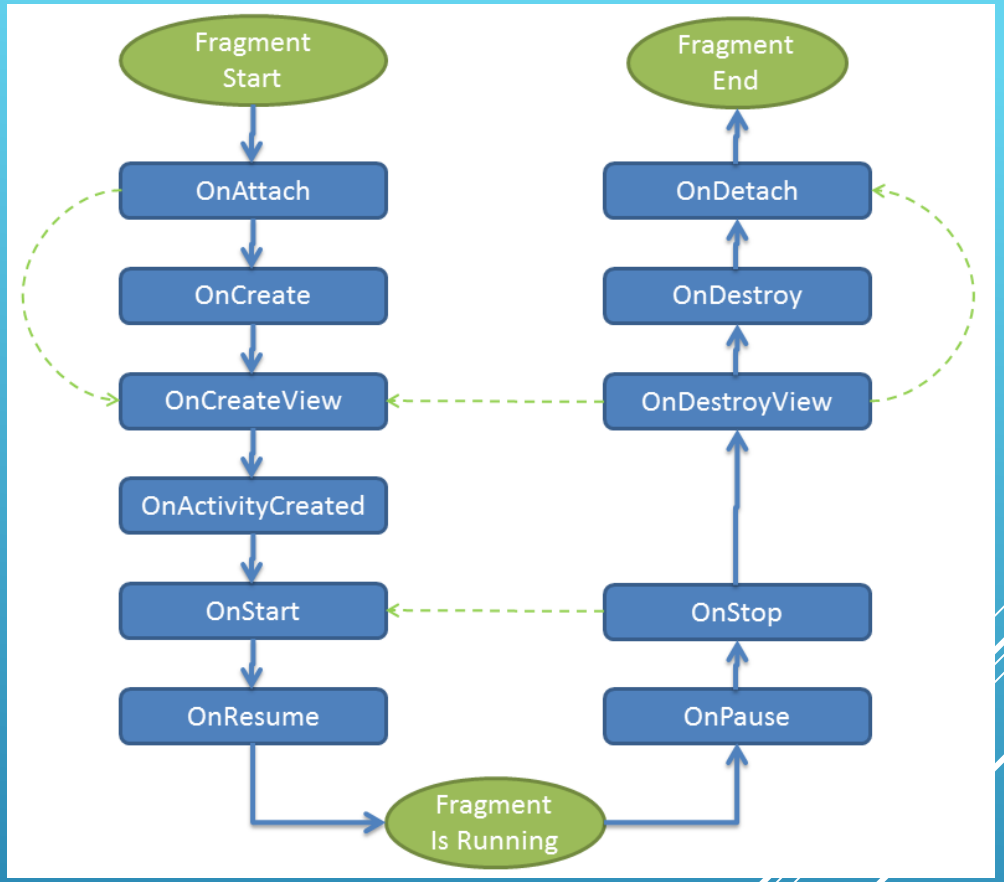
- Main Screen
- One displayed at a time
- Set the content using xml
- Handle user interactions



FRAGMENTS

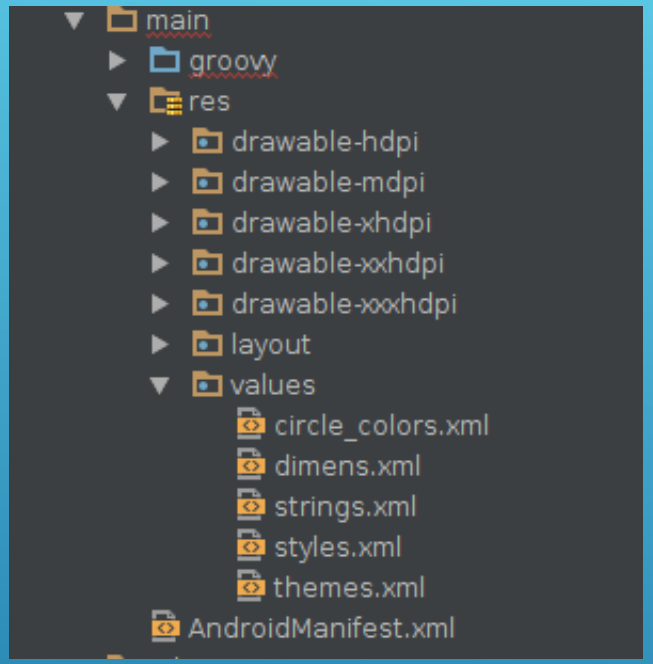
- Like activities
- Reusable UI components





VIEWS

- Component that is drawn on the screen
- Many easy to use provided ones for you (TextView, ImageView, etc)
- Can create custom ones (with drawing or composition).




```
contentView = R.layout.activity_main
```

RESOURCES

PITFALLS

- Slow (Use compile static)
- Dex Limit (use proguard)
- Android Plugin Updates a lot (breaks stuff)
- Groovy not fully supported by Android Studio

- SwissKnife
- Android Spock
- Grooid Tools

ANDROID AWESOMENESS

SWISSKNIFE

- Based off ButterKnife, Android Annotations, and Android AutoParcel
- Great AST Transformations
- Awesome Extension Methods

- @SaveInstance
- @OnClick, @OnItemClick
- @OnBackground, @OnUIThread
- @Parcelable
- @StringRes, @AnontationRes

SWISSKNIFE - ANNOTATIONS

```
class MyActivity extends Activity {

  @StringRes(R.string.important_message)
  String reallyImportantMessage

  @Extra("api_key")
  String apiKey

  @SaveInstance
  public String myString;

  @OnClick(R.id.button)
  public void onButtonClicked(Button button) {
    Toast.makeText(this, "Button clicked", Toast.LENGTH_SHOT).show();
  }

  @OnBackground
  public void doSomeProcessing(URL url) {
    // Contents will be executed on background
    ...
  }

  @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // This must be called for injection of views and callbacks to take place
        SwissKnife.inject(this);

        // This must be called for saved state restoring
        SwissKnife.restoreState(this, savedInstanceState);

        // This mus be called for automatic parsing of intent extras
        SwissKnife.loadExtras(this)
    }
}
```


```java
public final class Person implements Parcelable {
  private final int id
  private final String firstName
  private final String lastName

  public void writeToParcel(Parcel out, int flags) {
    out.writeInt(id);
    out.writeString(firstName);
    out.writeString(lastName);
  }

  public static final Parcelable.Creator<Person> CREATOR
    = new Parcelable.Creator<Person>() {
    public Person createFromParcel(Parcel in) {
      return new Person(in);
    }

    public Person[] newArray(int size) {
      return new Person[size];
    }
  };
```


```groovy
@Parcelable
class Person {
  int id
  String firstName
  String lastName
}
```

SPOCK ON ANDROID

- Extra classes for Android Specific Testing

```
@CompileStatic(TypeCheckingMode.SKIP)
protected View createView() {
    new AndroidBuilder().build(this) {
        relativeLayout(width: MATCH_PARENT, height: MATCH_PARENT, padding: [dp(64), dp(16)]) {
            textView(width: MATCH_PARENT, height: dp(20), text: R.string.hello_world)
        }
    } as View
}
```

GROOID TOOLS

EXAMPLE

QUESTIONS?