RESOURCES SUMMARY

How to work with resources

TRAINING ROADMAP: OVERVIEW

- Summary
- Custom views

This training covers best practices related to the UI view & resources.

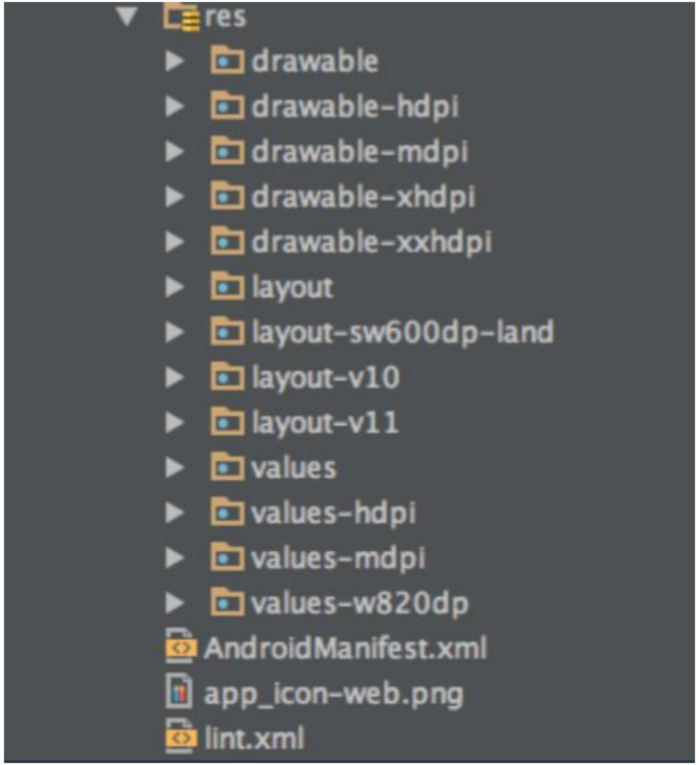
Pre-requisites:

- Readiness to learn new ☺

RESOURCES: SUMMARY

Android resources are one of the Android SDK powerful tools to work with the UI layer. Some common resources are:

- **Color**: **Colors** available for your application. In colors.xml you can declare your color palette.
- **Drawable**: Shapes and statics assets ready to be used in Image components as src or background.
- **Layout**: XML files to describe the application UI and to be inflated by the system.
- **Menu**: Android menus can be described in xml files and used directly from our activities.
- **Integer**: Integer values ready to be used in xml files or java code.
- **XML**: XML configuration files that can be used to indicate different
- configuration parameters. Not really common.



```
▼ res
    ▶ drawable
    ▶ drawable-hdpi
    ▶ drawable-mdpi
    ▶ drawable-xhdpi
    ▶ drawable-xxhdpi
    ▶ layout
    ▶ layout-sw600dp-land
    ▶ layout-v10
    ▶ layout-v11
    ▶ values
    ▶ values-hdpi
    ▶ values-mdpi
    ▶ values-w820dp
  AndroidManifest.xml
  app_icon-web.png
  lint.xml
```

RESOURCES: SUMMARY

- For complex projects where we have to support **tons of different devices** with different screen densities and sizes and different Android API versions, we need to use our resources and qualifiers properly to **get the max development performance.**
- **Qualifiers** are really important to work with different Android configurations. We can prepare different resources for different Android configurations and these resources are going to be provided automatically. C**onfiguration based on qualifiers:**
- **Screen density**: ldpi, mdpi, hdpi, xhdpi, etc.
- **Language**: ru, en, fr, en-us, etc.
- **Min width**: sw600dp, sw720dp.
- **Available width or height:** w720dp, h1024dp.
- **Screen orientation**: landscape, portrait.
- **Android API level**: v7, v8, v9, etc.

RESOURCES: SUMMARY

How have we used resources and qualifiers in Effective Android UI project?

- Layouts:
  - layout-v10: Android 2.X is using one layout with just one fragment.
  - layout-v11: Android 3.X y 4.X (except tablets in landscape).
  - layout-sw600dp-land: Layout with two fragments in landscape.
- Values:
  - values-mdpi: mdpi and ldpi devices are using one column for the GridView and different ImageView height.
  - values-hdpi: hdpi, xhdpi and xxhdpi devices with screen width lower than 820 dip are using two columns for the GridView and another ImageView height.
  - values-w820: Devices with more than 820 dip width are going to use 3 columns for the GridView configuration.

CUSTOM VIEWS

- **Main motivations** to use custom views are:
  - Create a **non implemented Android SDK widget**.
  - **Group presentation logic** for different views. Really useful with MVVM.
  - **Add semantic** to UI layer components.
  - **Avoid code duplicity.**

CUSTOM VIEWS

- There are 5 approaches to implement your custom views:
- **Extend an already written widget** like TextView, ImageView.
- **Extend View** and override **onDraw** method using the **Canvas** API.
- **Extend one Layout/ViewGroup** and inflate your own layout file.
- **Extend one Layout/ViewGroup** and wait for views added by other developers using this custom view.
- **Inflate different layouts** provided by the custom view client

RECOMMENDATION

- Avoid expensive tasks executed over the UI thread.
- Avoid code duplicity.
- Measure your UI performance using performance tools.
- Use themes, styles, dimensions and colors properly.
- Think in the UI layer like an isolated domain.
- Write testable code and test it.
- Return all the information needed to paint your UI.
- Implement your ListView and Adapters recycling your views.
- **Write code for your buddies, not for the machine.**