

Broadening Android App Appeal and Reach

WORKING WITH RESOURCES FOR LOCALIZATION AND ADAPTABILITY



Jim Wilson

MOBILE SOLUTIONS DEVELOPER & ARCHITECT

@hedgehogjim blog.jwhh.com



What to Expect from This Course



Adapt to Device Differences with Resources

Create Custom Views

Incorporate Accessibility Features

Deep Links and Synthesized Back Stack

Create and Update App Widgets



What to Expect from This Course



Adapt to Device Differences with Resources

Create Custom Views

Handle Custom View Sizing and Positioning

Make Custom Views Interactive & Configurable

Add Accessibility Features to Our Apps



What to Expect from This Module



Resources and Device Adaptability

Load Different Values Based on Screen Width

Create Orientation-specific Layouts

Localize UI Text into Alternate Languages

Use Different Drawables Based Screen Density

Resource Folders Physical Structure



Resources and Adaptability

Resources externalize content from code

- Maintained separately from source code
- Simplify adaptability

Android supports a rich device ecosystem

- Apps need to adapt to device configuration differences



Resources and Adaptability

Hardware differences

- Screen size
- Screen density

Dynamic configuration differences

- Portrait or landscape orientation
- Night mode on or off

Setup and installation differences

- Locale
- Installed version of Android



Resources and Adaptability

Dealing with differences can be challenging

- Manual solutions are code intensive

Android resources handle differences

- Can be associated with configurations
- Android handles selected appropriate resource for current configuration



Drawable Resources

Something that can be drawn to the screen

- Images for menus, navigation drawer, status bar, notifications, etc.

Vector

- Scalable vector graphics
- Work well with simple graphics

Raster

- Non-scalable graphics
- *.png preferred, can also use *.jpg, *.gif



Drawable Resources

Devices have variety of pixel densities

- Presents challenge for raster graphics

Raster graphics don't scale well

- Stretch when scaled to higher densities



Need different images for each density



Drawable Resources

Devices categorized by screen density

- Separate drawable folder for each
- Create graphic files sized appropriately for each screen density



Drawable Resources

| Category | Pixels Per Inch | Drawable Folder |
|----------|-----------------|------------------|
| mdpi | ~160 | drawable-mdpi |
| hdpi | ~240 | drawable-hdpi |
| xhdpi | ~320 | drawable-xhdpi |
| xxhdpi | ~480 | drawable-xxhdpi |
| xxxhdpi | ~640 | drawable-xxxhdpi |



Summary



Resources externalize content from code

- Simplify adaptability

Android supports rich device ecosystem

- Broad set of device characteristics

Can associate resources with characteristics

- Android handles details of selecting appropriate resource for device



Summary



Some characteristics can change at runtime

- Android destroys activity
- Recreates with appropriate resources

Characteristic specific layouts

- Device orientation, screen size, etc.
- Be sure to keep view IDs consistent

Summary



Use string resources for UI text

- Simplifies multi-language support
- Devices have a language specific locale
- Can associate string resources with specific languages



Summary



Include density specific drawables

- Raster graphics don't scale well

Characteristic specific resource folders

- Visible in “Project” view of Android Studio project window

