ITW202: Mobile Application

Unit IV: Developing for Android

Ms. Sonam Wangmo

Gyalpozhing College of Information Technology
Royal University of Bhutan

May 16, 2021

Adaptive layouts and resources

What are adaptive layouts?

- An adaptive layout is a layout that works well for different screen sizes and orientations, different devices, different locales and languages, and different versions of Android.

How to create an adaptive layout

1. Externalizing Resources.
2. Grouping Resources
3. Providing Alternative Resources
4. Provind default Resources

Externalizing Resources

- When you externalize resources, you keep them separate from your app code.
- For example, instead of hard-coding a string into your code, you name the string and add it to the strings.xml file

Externalizing Resources

Always externalize resources such as drawables, icons, layouts, and strings. Here's why it's important:

- Firstly, You can maintain externalized resources separately from your other code. If a resource is used in several places in your code and you need to change the resource, you only need to change it in one place.

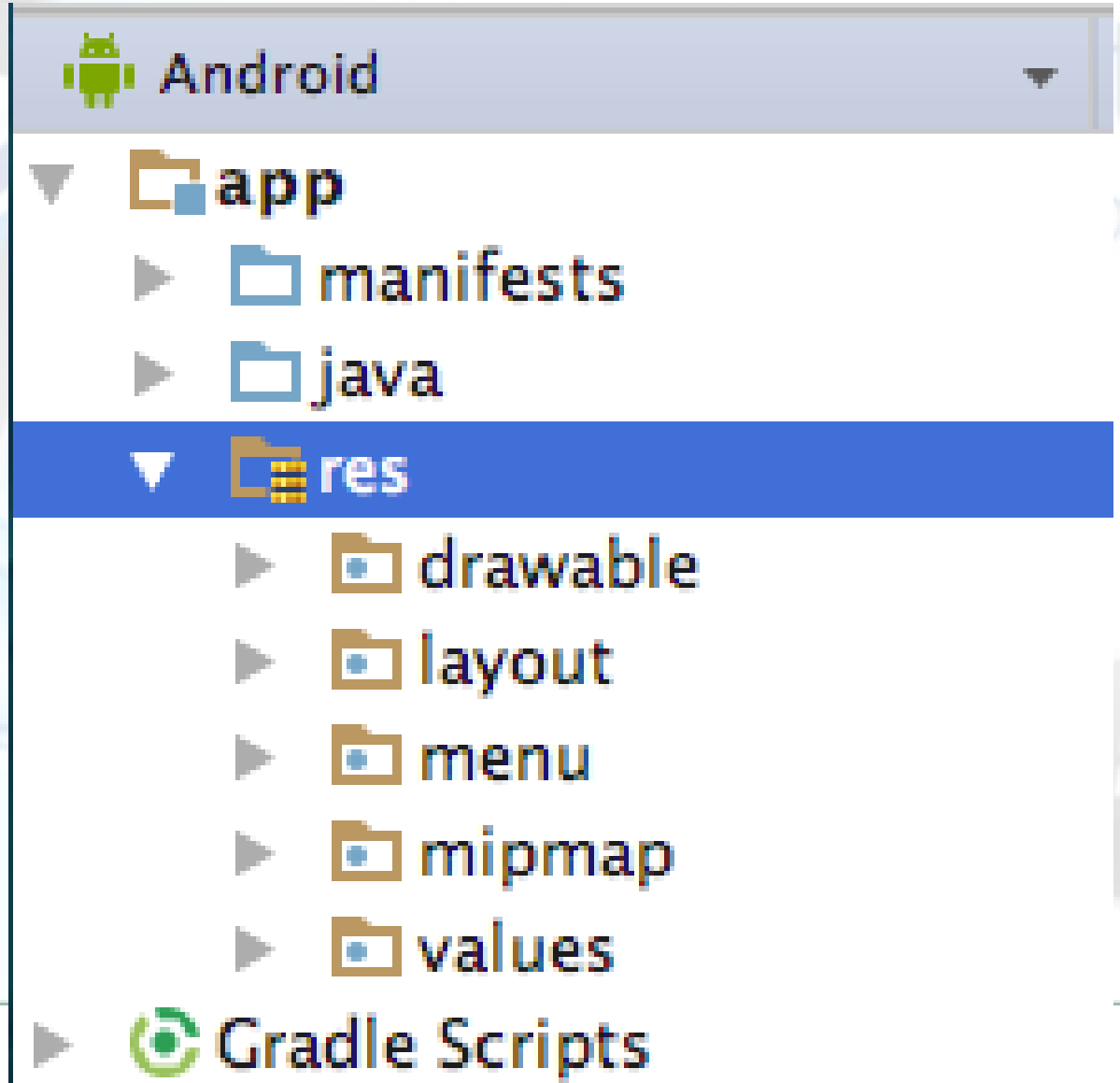
Externalizing Resources

- Secondly, You can provide alternative resources that support specific device configurations, for example devices with different languages or screen sizes. This becomes increasingly important as more Android-powered devices become available.

Grouping Resources

- Store all your resources in the res folder. Organize resources by type into folders within res. You must use standardized names for these folders.

Grouping Resources



Alternative Resources

- Most apps provide alternative resources to support specific device configurations.
- For example, your app should include alternative drawable resources for different screen densities, and alternative string resources for different languages.

Alternative Resources

- Like default resources, alternative resources are kept in folders inside res.
- Alternative-resource folders use the following naming convention:

  resource_name-config_qualifier

Alternative Resources

- resource_name: is the folder name for this type of resource. For example, drawable or values.
- config_qualifier: Specifies a device configuration for which these resources are used.

Alternative Resources

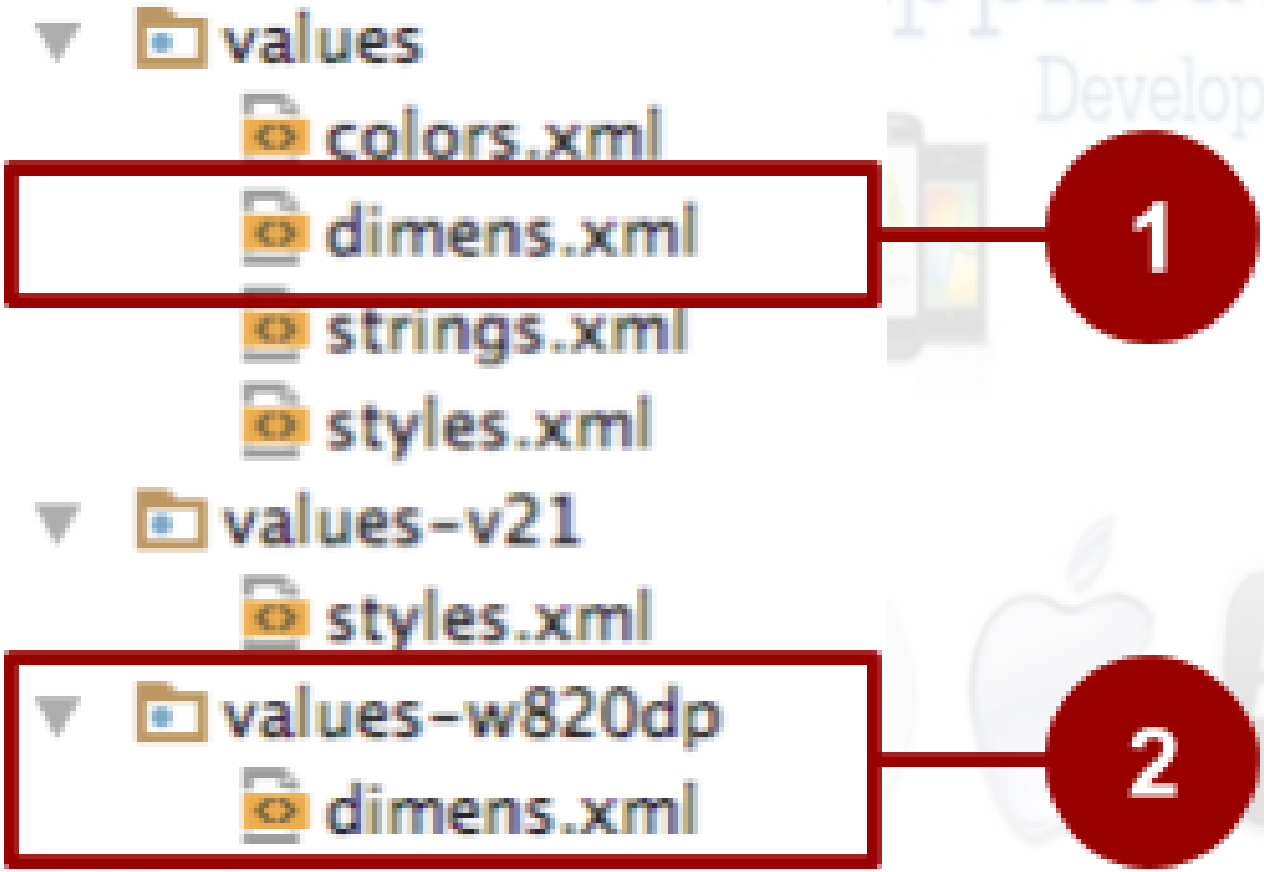
- To add multiple qualifiers to one folder name, separate the qualifiers with a dash. If you use multiple qualifiers for a resource folder, you must list them in the order they are listed in the table.

Alternative Resources

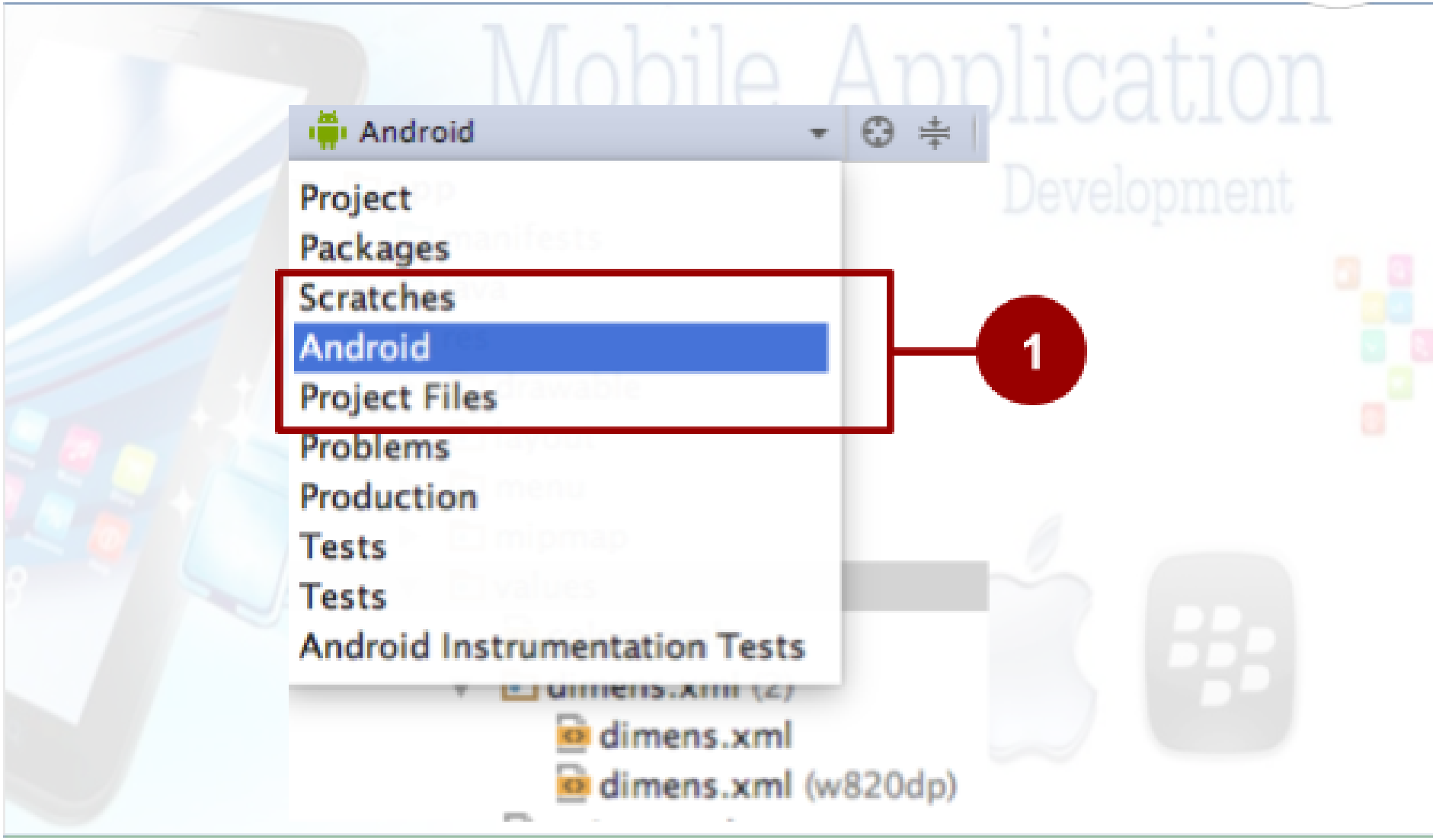
Example: String resources localized to Japanese would be in a strings.xml file inside the values-ja folder in the res folder (abbreviated as res/values-ja/strings.xml).

Example: Style resources for API level 21 and higher would be in a res/values-v21/styles.xml file.

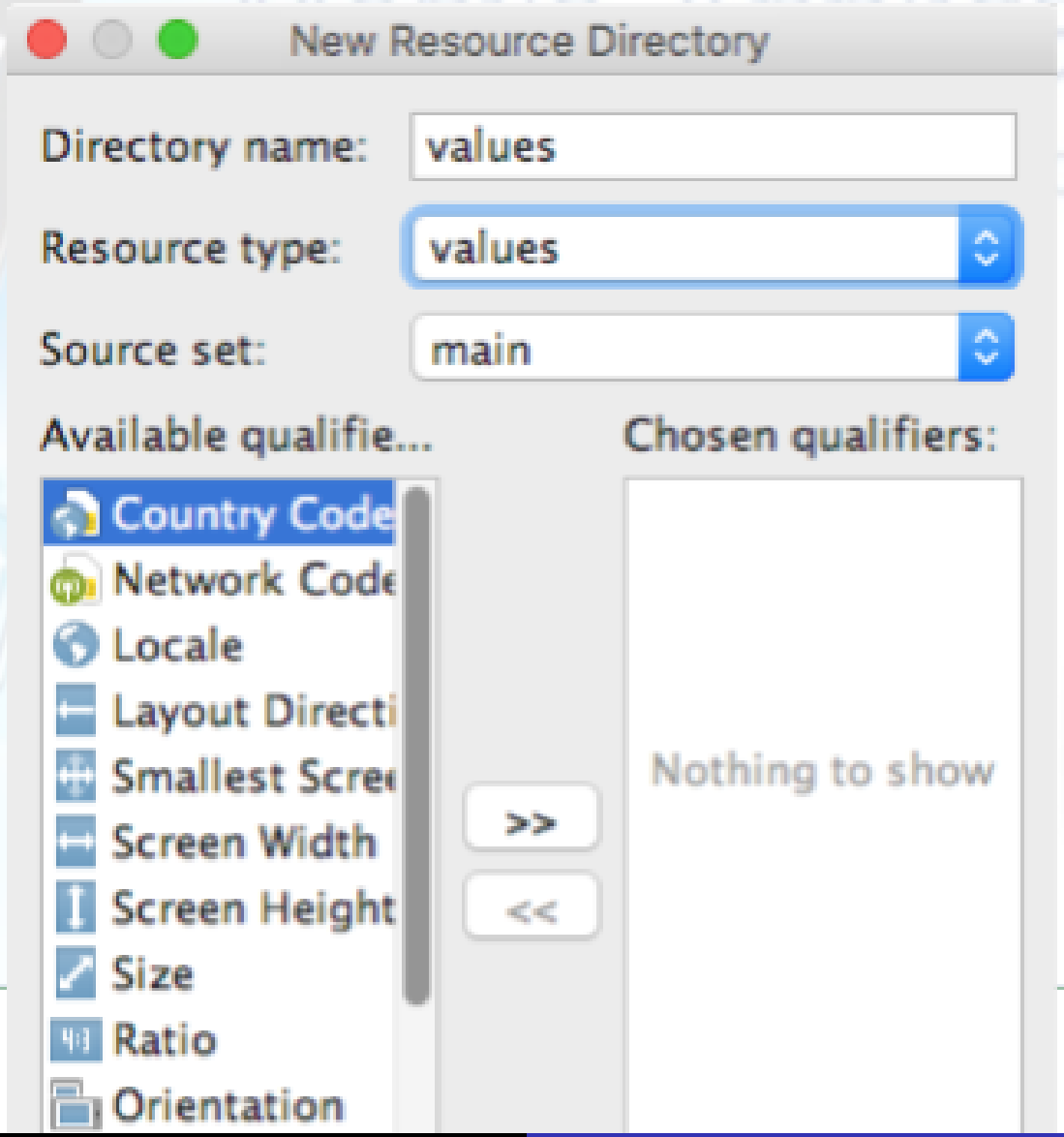
Alternative Resources



Creating alternative resources



Creating alternative resources



Common alternative-resource qualifiers

Screen orientation

The screen-orientation qualifier has two possible values:

- port: The device is in portrait mode (vertical). For example, res/layout-port/
- land: The device is in landscape mode (horizontal). For example, res/layout-land/

Common alternative-resource qualifiers

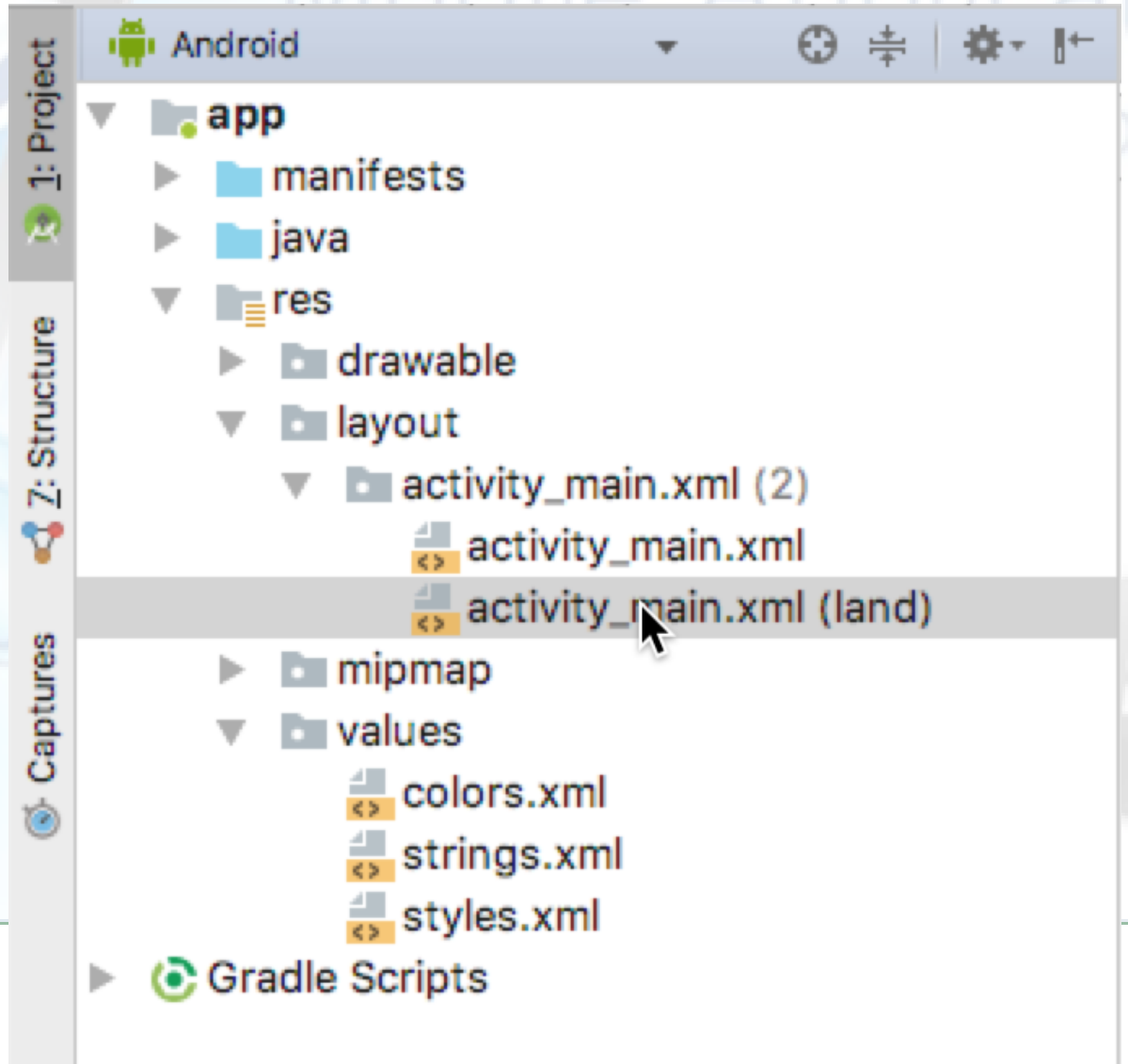
Screen orientation

To create variants of your layout XML file for landscape orientation and larger displays, use the layout editor. To use the layout editor:

1. In Android Studio, open the XML file (such as activity_main.xml). The layout editor appears.
2. Click the Design tab at the bottom of the layout editor (if it is not already selected).
3. Click the Orientation in Editor button
4. Choose an option such as Create Landscape Variation.

Common alternative-resource qualifiers

Screen orientation



Common alternative-resource qualifiers

Smallest width

- The smallest-width qualifier, specifies the minimum width of the device.
- It is the shortest of the screen's available height and width, the "smallest possible width" for the screen.
- Specify smallest width in dp units, using the following format:

  swndp, where n is the minimum width.

Common alternative-resource qualifiers

Smallest width

Example: Resources in a file named res/values-sw320dp/styles.xml are used if the device's screen is always at least 320dp wide.

Common alternative-resource qualifiers

Smallest width

Some values for common screen sizes:

- 320, for devices with screen configurations such as 240x320 ldpi (QVGA handset), 320x480 mdpi (handset), and 480x800 hdpi (high-density handset)
- 480, for screens such as 480x800 mdpi (tablet/handset)
- 600, for screens such as 600x1024 mdpi (7" tablet)
- 720, for screens such as 720x1280 mdpi (10" tablet)

Common alternative-resource qualifiers

Smallest width

Android uses resource closest to (without exceeding) the device's smallest width

Common alternative-resource qualifiers

Platform Version

- The platform-version qualifier specifies the minimum API level supported by the device.

  For example, use v11 for API level 11 (devices with Android 3.0 or higher).

Common alternative-resource qualifiers

Platform Version

- Use the platform-version qualifier when you use resources for functionality that's unavailable in prior versions of Android.
- For example: WebP image format requires API level 14 (Android 4.0)

Common alternative-resource qualifiers

Localization

- The localization qualifier specifies a language and, optionally, a region.
- Increases potential audience for your app

Common alternative-resource qualifiers

Localization

Example: If the user changes the language or region in the device's system settings while your app is running, and if alternative resources are available, Android automatically reloads your app with alternative resources that match the new device configuration.

Providing default resources

- Default resources specify the default design and content for your application.
- Always provide default resources
  - directory name without a qualifier
  - res/layout, res/values, res/drawables. . . .
- Android falls back on default resources when no specific resources match configuration.


Mobile Application
Development
THANK YOU