Android Developers

Maintaining Compatibility

This lesson teaches you to

Define Alternative Styles

Provide Alternative Layouts

Use the Support Library

Check the System Version

You should also read

Material design specification

Material design on Android

Some material design features like the material theme and custom activity transitions are only available on Android 5.0 (API level 21) and above. However, you can design your apps to make use of these features when running on devices that support material design and still be compatible with devices running previous releases of Android.

Define Alternative Styles

You can configure your app to use the material theme on devices that support it and revert to an older theme on devices running earlier versions of Android:

1. Define a theme that inherits from an older theme (like Holo) in `res/values/styles.xml`.

2. Define a theme with the same name that inherits from the material theme in `res/values-v21/styles.xml`.

3. Set this theme as your app's theme in the manifest file.

> **Note:** If your app uses the material theme but does not provide an alternative theme in this manner, your app will not run on versions of Android earlier than 5.0.

This site uses cookies to store your preferences for site-specific language and display options.

OK

If the layouts that you design according to the material design guidelines do not use any of the new XML attributes introduced in Android 5.0 (API level 21), they will work on previous versions of Android. Otherwise, you can provide alternative layouts. You can also provide alternative layouts to customize how your app looks on earlier versions of Android.

Create your layout files for Android 5.0 (API level 21) inside `res/layout-v21/` and your alternative layout files for earlier versions of Android inside `res/layout/`. For example, `res/layout/my_activity.xml` is an alternative layout for `res/layout-v21/my_activity.xml`.

To avoid duplication of code, define your styles inside `res/values/`, modify the styles in `res/values-v21/` for the new APIs, and use style inheritance, defining base styles in `res/values/` and inheriting from those in `res/values-v21/`.

Use the Support Library

The v7 Support Libraries (https://developer.android.com/tools/support-library/features.html#v7) r21 and above includes the following material design features:

- Material design styles (https://developer.android.com/training/material/theme.html) for some system widgets when you apply one of the `Theme.AppCompat` themes.

- Color palette theme attributes (https://developer.android.com/training/material/theme.html#ColorPalette) in the `Theme.AppCompat` themes.

- The `RecyclerView` (https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html) widget to display data collections (https://developer.android.com/training/material/lists-cards.html#RecyclerView).

- The `CardView` (https://developer.android.com/reference/android/support/v7/widget/CardView.html) widget to create cards (https://developer.android.com/training/material/lists-cards.html#CardView).

- The `Palette` (https://developer.android.com/reference/android/support/v7/graphics/Palette.html) class to extract prominent colors from images (https://developer.android.com/training/material/drawables.html#ColorExtract).

Updated support library versions may provide additional support for material design features. See the Support Library revision history (https://developer.android.com/topic/libraries/support-library/revisions.html) for more information.

System widgets

The `Theme.AppCompat` themes provide material design styles for these widgets:

- `EditText` (https://developer.android.com/reference/android/widget/EditText.html)

- `Spinner` (https://developer.android.com/reference/android/widget/Spinner.html)

This site uses cookies to store your preferences for site-specific language and display options.

  OK

- SwitchCompat (https://developer.android.com/reference/android/support/v7/widget/SwitchCompat.html)

- CheckedTextView (https://developer.android.com/reference/android/widget/CheckedTextView.html)

Color Palette

To obtain material design styles and customize the color palette with the Android v7 Support Library, apply one of the `Theme.AppCompat` themes:

```
<!-- extend one of the Theme.AppCompat themes -->
<style name="Theme.MyTheme" parent="Theme.AppCompat.Light">
    <!-- customize the color palette -->
    <item name="colorPrimary">@color/material_blue_500</item>
    <item name="colorPrimaryDark">@color/material_blue_700</item>
    <item name="colorAccent">@color/material_green_A200</item>
</style>
```

Lists and Cards

The RecyclerView (https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html) and CardView (https://developer.android.com/reference/android/support/v7/widget/CardView.html) widgets are available in earlier versions of Android through the Android v7 Support Library with these limitations:

- CardView (https://developer.android.com/reference/android/support/v7/widget/CardView.html) falls back to a programmatic shadow implementation using additional padding.

- CardView (https://developer.android.com/reference/android/support/v7/widget/CardView.html) does not clip its children views that intersect with rounded corners.

Dependencies

To use these features in versions of Android earlier than 5.0 (API level 21), include the Android v7 Support Library in your project as a Gradle dependency (https://developer.android.com/studio/build/index.html#dependencies):

```
dependencies {
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support:cardview-v7:25.3.1'
    compile 'com.android.support:recyclerview-v7:25.3.1'
}
```

Check the System Version

The following framework features are available only in Android 5.0 (API level 21) and above:

This site uses cookies to store your preferences for site-specific language and display options.

OK

- Reveal animations

- Path-based animations

To preserve compatibility with earlier versions of Android, check the system version
(https://developer.android.com/reference/android/os/Build.VERSION.html#SDK_INT) at runtime before you invoke the APIs
for any of these features:

```
// Check if we're running on Android 5.0 or higher
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
    // Call some material design APIs here
} else {
    // Implement this feature without material design
}
```

> **Note:** To specify which versions of Android your app supports, use the `android:minSdkVersion` and `android:targetSdkVersion` attributes in your manifest file. To use the material design features in Android 5.0, set the `android:targetSdkVersion` attribute to `21`. For more information, see the <uses-sdk> API guide (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html).

This site uses cookies to store your preferences for site-specific language and display options.

OK