

# UI Overview

All user interface elements in an Android app are built using **View**

(<https://developer.android.com/reference/android/view/View.html>) and **ViewGroup**

(<https://developer.android.com/reference/android/view/ViewGroup.html>) objects. A **View**

(<https://developer.android.com/reference/android/view/View.html>) is an object that draws something on the screen that the user can interact with. A **ViewGroup**

(<https://developer.android.com/reference/android/view/ViewGroup.html>) is an object that holds other **View**

(<https://developer.android.com/reference/android/view/View.html>) (and **ViewGroup**

(<https://developer.android.com/reference/android/view/ViewGroup.html>)) objects in order to define the layout of the interface.

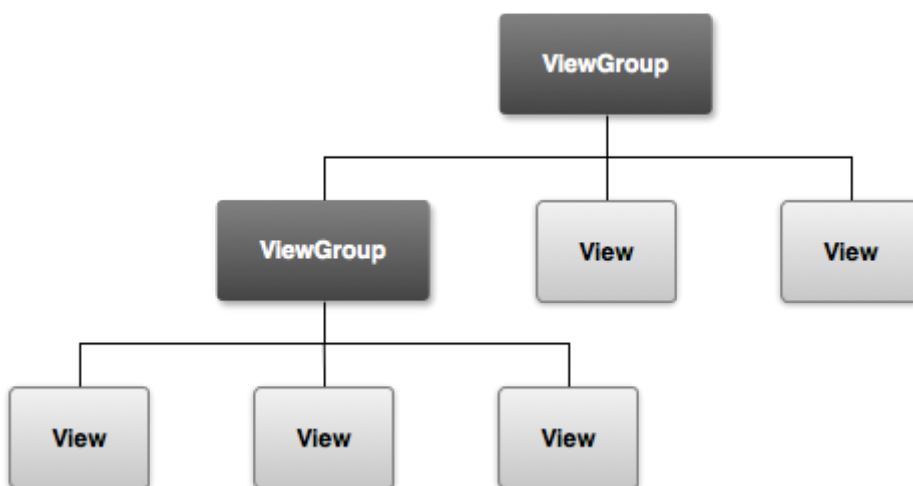
Android provides a collection of both **View** (<https://developer.android.com/reference/android/view/View.html>) and **ViewGroup** (<https://developer.android.com/reference/android/view/ViewGroup.html>) subclasses that offer you common input controls (such as buttons and text fields) and various layout models (such as a linear or relative layout).

## User Interface Layout

The user interface for each component of your app is defined using a hierarchy of **View**

(<https://developer.android.com/reference/android/view/View.html>) and **ViewGroup**

(<https://developer.android.com/reference/android/view/ViewGroup.html>) objects, as shown in figure 1. Each view group is an invisible container that organizes child views, while the child views may be input controls or other widgets that draw some part of the UI. This hierarchy tree can be as simple or complex as you need it to be (but simplicity is best for performance).



**Figure 1.** Illustration of a view hierarchy, which defines a UI layout.

To declare your layout, you can instantiate [View](https://developer.android.com/reference/android/view/View.html) objects in code and start building a tree, but the easiest and most effective way to define your layout is with an XML file. XML offers a human-readable structure for the layout, similar to HTML.

The name of an XML element for a view is respective to the Android class it represents. So a `<TextView>` element creates a [TextView](https://developer.android.com/reference/android/widget/TextView.html) widget in your UI, and a `<LinearLayout>` element creates a [LinearLayout](https://developer.android.com/reference/android/widget/LinearLayout.html) view group.

For example, a simple vertical layout with a text view and a button looks like this:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a Button" />
</LinearLayout>
```

When you load a layout resource in your app, Android initializes each node of the layout into a runtime object you can use to define additional behaviors, query the object state, or modify the layout.

For a complete guide to creating a UI layout, see [XML Layouts](https://developer.android.com/guide/topics/ui/declaring-layout.html).

## User Interface Components

You don't have to build all of your UI using [View](https://developer.android.com/reference/android/view/View.html) and [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup.html) objects. Android provides several app components that offer a standard UI layout for which you simply need to define the content. These UI components each have a unique set of APIs that are described in their respective documents, such as [Adding the App Bar](https://developer.android.com/training/appbar/index.html), [Dialogs](https://developer.android.com/guide/topics/ui/dialogs.html), and [Status Notifications](https://developer.android.com/guide/topics/ui/notifiers/notifications.html).