**ANDROID NOTIFICATIONS**

A notification is a message you can display to the user outside of your application's normal UI. When you tell the system to issue a notification, it first appears as an icon in the **notification area**. To see the details of the notification, the user opens the **notification drawer**. Both the notification area and the notification drawer are system-controlled areas that the user can view at any time.

* **Notification Display Elements**

Notifications in the notification drawer can appear in one of two visual styles, depending on the version and the state of the drawer:

Normal view

The standard view of the notifications in the notification drawer.

Big view

A large view that's visible when the notification is expanded. Big view is part of the expanded notification feature available as of Android 4.1.

1. **Normal View:**

A notification in normal view appears in an area that's up to 64 dp tall. Even if you create a notification with a big view style, it will appear in normal view until it's expanded. This is an example of a normal view:



1. Content Title
2. Large Icon
3. Content Text
4. Content Info
5. Small Icon
6. Time at which notification was issued. You can set an explicit value with [setWhen()](https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#setWhen%28long%29); if you don't it defaults to the time that the system received the notification.
7. **Big View:**

A notification's big view appears only when the notification is expanded, which happens when the notification is at the top of the notification drawer, or when the user expands the notification with a gesture. Expanded notifications are available starting with Android 4.1.

The following screenshot shows an inbox-style notification:



The elements of big view are similar to normal view **except element number 7** i.e. the details area**.**

Big picture style

The details area contains a bitmap up to 256 dp tall in its detail section.

Big text style

Displays a large text block in the details section.

Inbox style

Displays lines of text in the details section.

All of the big view styles also have the following content options that aren't available in normal view:

Big content title

Allows you to override the normal view's content title with a title that appears only in the expanded view.

Summary text

Allows you to add a line of text below the details area.

Applying a big view style to a notification is described in the section [Applying a big view style to a notification](https://developer.android.com/guide/topics/ui/notifiers/notifications.html#ApplyStyle).

* **Creating a Notification:**

You can specify notification UI information and actions through NotificationCombat.Builder object. To create the notification itself, you call [NotificationCompat.Builder.build()](https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#build%28%29), which returns a [Notification](https://developer.android.com/reference/android/app/Notification.html) object containing your specifications. To issue the notification, you pass the [Notification](https://developer.android.com/reference/android/app/Notification.html) object to the system by calling [NotificationManager.notify()](https://developer.android.com/reference/java/lang/Object.html#notify%28%29).

**Required notification contents:**

1. A small Icon(setSmallIcon())
2. Content Title(setContentTitle())
3. Content Text(setContentText())

**Notification Actions:**

Although they are optional, you should add at least one action to your Android notification. An action allows user to go directly from notification to an activity in your application.

Inside a [Notification](https://developer.android.com/reference/android/app/Notification.html), the action itself is defined by a [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent.html) containing an [Intent](https://developer.android.com/reference/android/content/Intent.html) that starts an [Activity](https://developer.android.com/reference/android/app/Activity.html) in your application. To associate the [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent.html) with a gesture, call the appropriate method of [NotificationCompat.Builder](https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html). For example, if you want to start [Activity](https://developer.android.com/reference/android/app/Activity.html) when the user clicks the notification text in the notification drawer, you add the [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent.html) by calling [setContentIntent()](https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#setContentIntent%28android.app.PendingIntent%29).

**Code for creating a simple notification(Normal View):**

// Creates an explicit intent for an Activity in your app

**Intent resultIntent = new Intent(this, ResultActivity.class);**  
// The stack builder object will contain an artificial back stack for the  
// started Activity.  
// This ensures that navigating backward from the Activity leads out of  
// your application to the Home screen.

**TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);**

// Adds the back stack for the Intent (but not the Intent itself)

**stackBuilder.addParentStack(ResultActivity.class);**

// Adds the Intent that starts the Activity to the top of the stack

**stackBuilder.addNextIntent(resultIntent);  
PendingIntent resultPendingIntent =  
        stackBuilder.getPendingIntent(  
            0,  
            PendingIntent.FLAG\_UPDATE\_CURRENT  
        );**

**NotificationCompat.Builder mBuilder =  
        new NotificationCompat.Builder(this)  
        .setSmallIcon(R.drawable.notification\_icon)  
        .setContentTitle("My notification")  
        .setContentText("Hello World!");**

**mBuilder.setContentIntent(resultPendingIntent);**

**NotificationManager mNotificationManager =  
    (NotificationManager) getSystemService(Context.NOTIFICATION\_SERVICE);**// mId allows you to update the notification later on.  
**mNotificationManager.notify(mId, mBuilder.build());**