

Android Notification

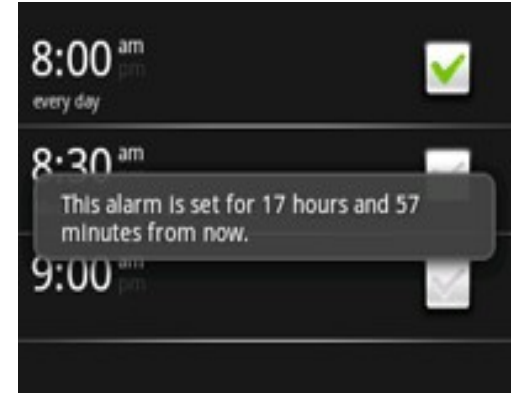
Topics

- Toast notification
- Status bar notification
- Dialog notification (we already covered this)
- Alarm manager

Toast Notification

Toast Notification

- A toast notification is a message that pops up on the surface of the window.
- It only fills the amount of space required for the message and the user's current activity remains visible and interactive.
- The notification automatically fades in and out, and does not accept interaction events.
- Because a toast can be created from a background Service, it appears even if the application isn't visible.



Toast Message

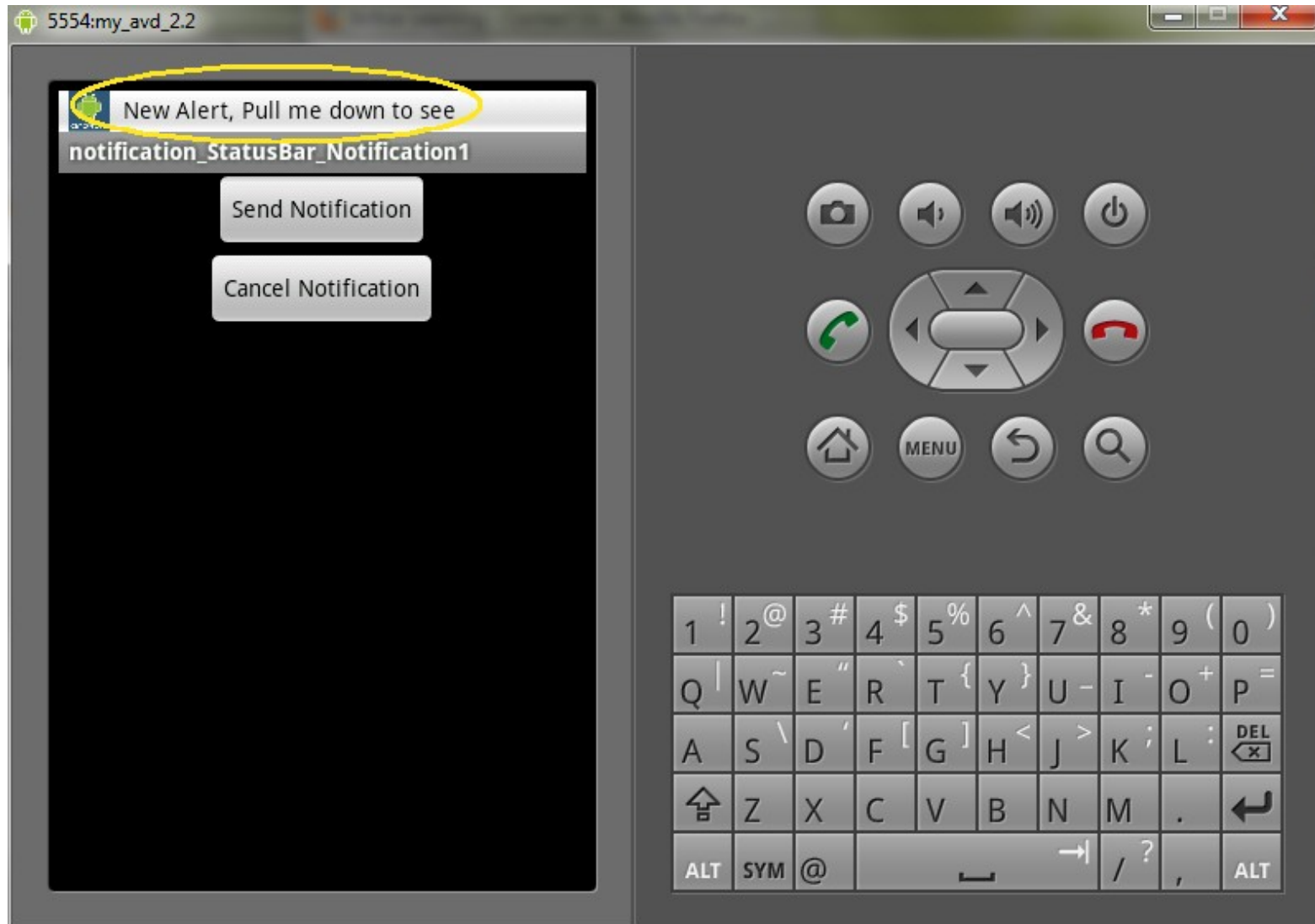
```
Toast.makeText(  
    getApplicationContext(),      // Context  
    R.string.toast_message,        // Get string resource to display  
    Toast.LENGTH_LONG).show();    // Make sure you call show()
```

Status Bar Notification

Status Bar Notification

- A status bar notification adds an icon to the system's status bar (with an optional ticker-text message) and an expanded message (notification detail) in the "Notifications" window.
- When the user selects the expanded message, Android fires an Intent that is defined by the notification (usually to launch an Activity).
- You can also configure the notification to alert the user with a sound, a vibration, and flashing lights on the device.

Icon and Ticker-text Message



Code that Displays Icon/Ticker-text

```
// Get a reference to the NotificationManager:
NotificationManager mNotificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);

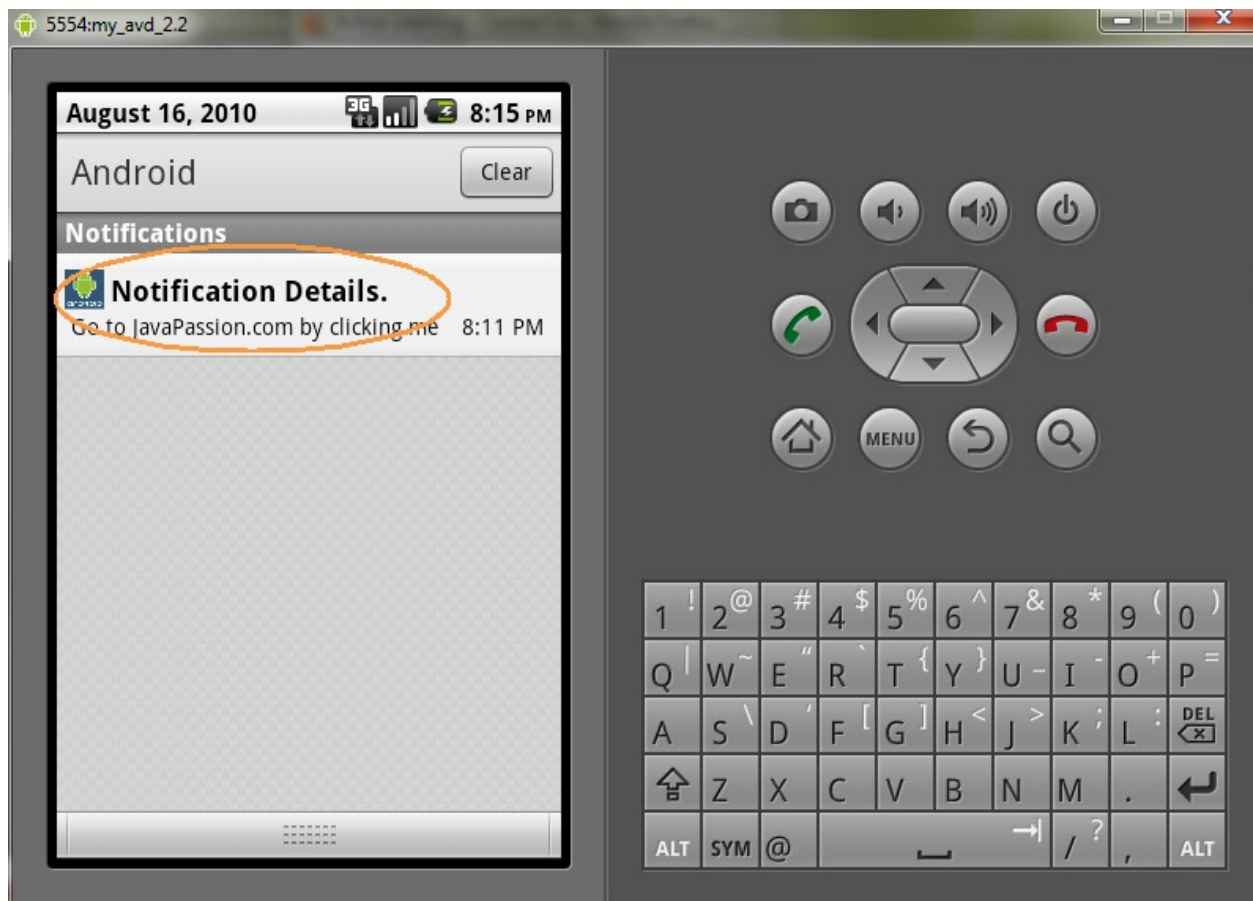
// Instantiate the Notification.
Notification mNotification = new Notification(R.drawable.android,
        "New Alert, Pull me down to see Notification details!",
        System.currentTimeMillis());

....
...

mNotificationManager.notify(SIMPLE_NOTIFICATION_ID,
        mNotification);
```

Expanded Message (Notification Detail)

- Appears when icon/ticker-text is pulled down



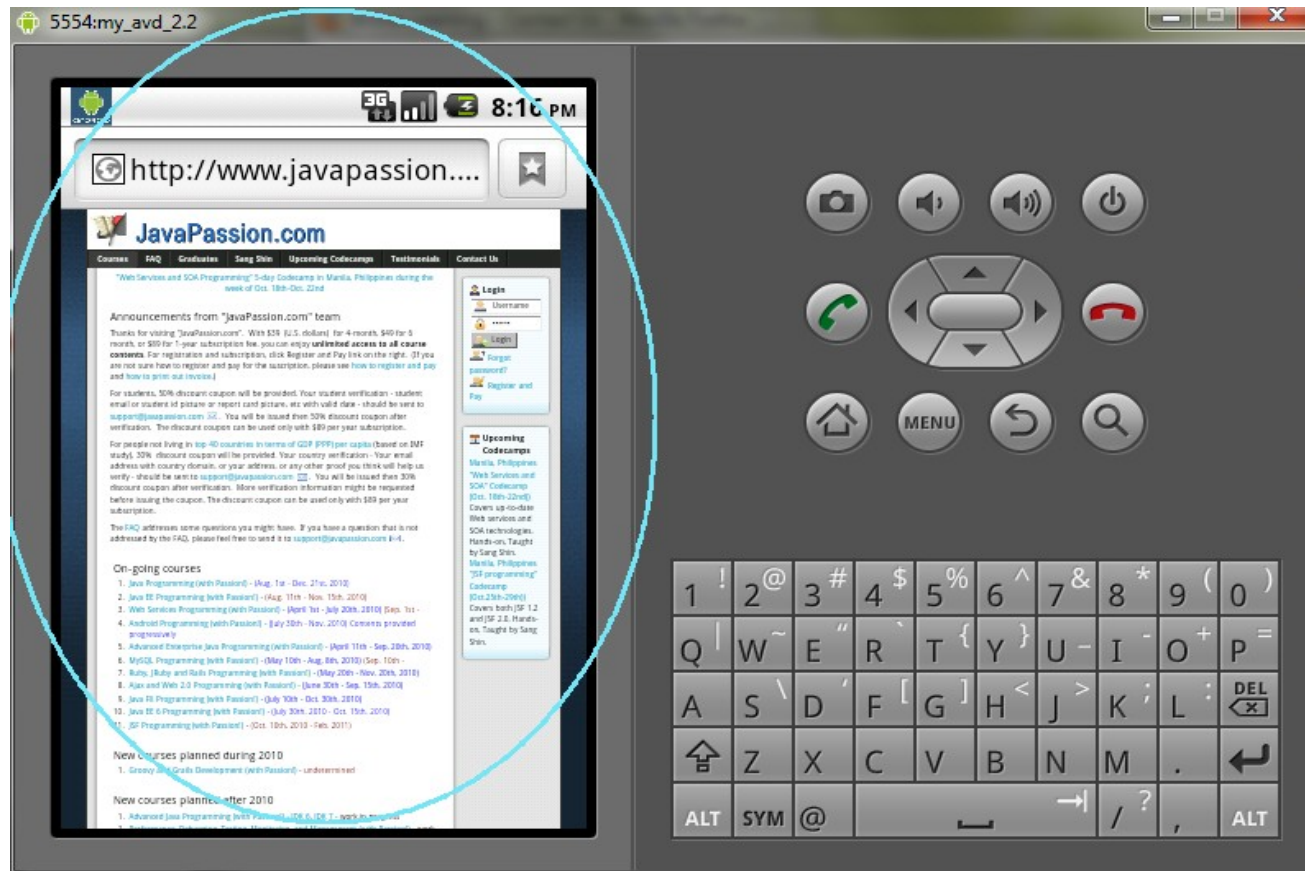
Code that Displays Expanded Message (Notification Detail)

```
// Define the Notification's expanded message (notification detail)
// and Intent. The notification detail message is the one that gets
// displayed when a user drags the notification downward.
CharSequence contentTitle = "Notification Details.";
CharSequence contentText = "Go to JavaPassion.com by clicking me";

....
...

// Sets the contentView field to be a view with the standard "Latest Event"
// layout. "mPendingIntent" is an intent to launch when the user clicks
// the expanded notification
mNotification.setLatestEventInfo(getApplicationContext(),
                                contentTitle,
                                contentText,
                                mPendingIntent);
```

Activity That Gets Started When Notification Detail is Clicked



Code that Starts an Activity when Notification Detail is clicked

```
// The Intent is to define an action that gets executed when a
// user clicks the notification detail.
intent notifyIntent = new Intent(
    android.content.Intent.ACTION_VIEW, Uri
        .parse("http://www.javapassion.com"));

// The PendingIntent can be handed to other applications so that they can
// perform the action you described on your behalf at a later time.
// By giving a PendingIntent to another application, you are granting it the
// right to perform the operation you have specified as if the other application
// was yourself (with the same permissions and identity).
PendingIntent mPendingIntent = PendingIntent.getActivity(
    StatusBarNotification.this, 0, notifyIntent,
    android.content.Intent.FLAG_ACTIVITY_NEW_TASK);

// Sets the contentView field to be a view with the standard "Latest Event"
// layout. "mPendingIntent" is an intent to launch when the user clicks
// the expanded notification
mNotification.setLatestEventInfo(getApplicationContext(),
    contentTitle,
    contentText,
    mPendingIntent);
```

Alarm Manager Example

Thank you