

Notifications

Chapter - 7

Prepared By : Dharmendra Ambani (Harivandana College – Rajkot)

Notify The User

- A notification is a user interface element that you display outside your app's normal UI to indicate that an event has occurred.

Notifying with the status bar

Typically the status bar shows information such as the current date and time.

It also displays notification as they arrive.

A simple notification has a number of important components:

- An icon
- Ticker Text
- Notification title text
- Notification body text

Building a Notification

- PendingIntent p = PendingIntent.getActivity(**this**, 0, **new Intent(MainActivity.this, MainActivity.class)**, 0, null);
-
- Notification notification = **new NotificationCompat.Builder(this)**
- .setTicker("Ticker")
- .setSmallIcon(android.R.drawable.arrow_up_float)
- .setContentTitle("Title")
- .setContentText("Text")
- .setContentIntent(p)
- .setAutoCancel(**true**)
- .build();
-
- NotificationManager n =
(NotificationManager) getSystemService(NOTIFICATION_SERVICE);
-
- n.notify(0, notification);

Modify(Updating) a Notifications

- To set up a Notification so it can be updated,, issue it with a Notification ID by calling `NotificationManager.Notify(ID,notification)`.

Vibrate the Phone

- ◉ Making Android phones vibrate is a good way to provide feedback to users or to interact with users even when phone volume is low.
- ◉ Entry on Manifest file
- ◉ `<uses-permission
android:name="android.permission.VIBRATE">`

Blinking the Lights

- Blinking lights are a great way to pass information silently to the user when other forms of alert are not appropriate.
- `.setLights(Color.Green,1,1);`
- 1 = On Milliseconds
- 1 = Off Milliseconds

Customizing the Notification

- Although the default notification behavior in the expanded status bar tray is sufficient for most purpose, developers can customize how notifications are displayed if they so choose using RemoteViews class with Layouts.

● End