Notification Manager

Tushar B. Kute,

http://tusharkute.com

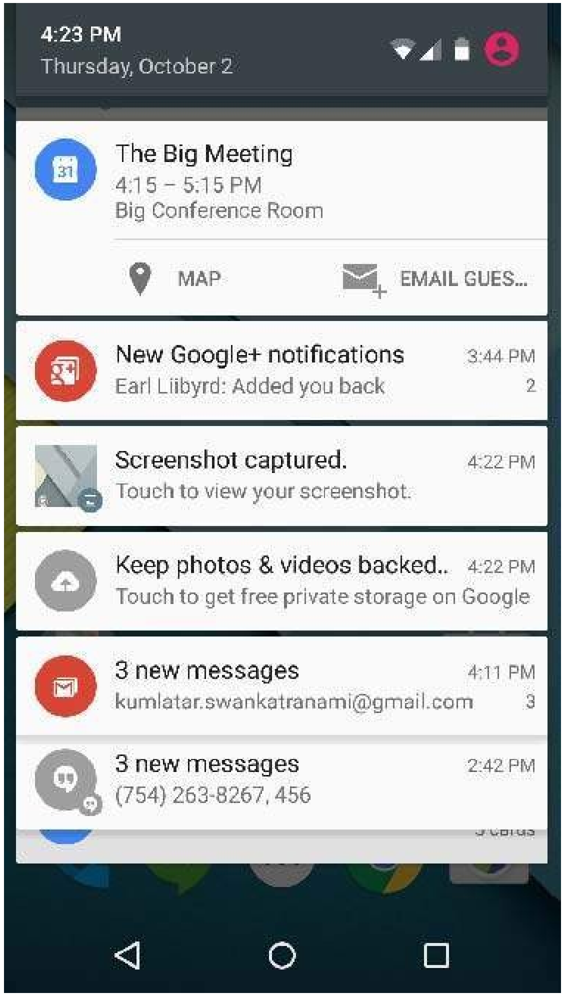
Notification

- 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

- Android Toast class provides a handy way to show users alerts but problem is that these alerts are not persistent which means alert flashes on the screen for a few seconds and then disappears.
- To see the details of the notification, you will have to select the icon which will display notification drawer having detail about the notification. While working with emulator with virtual device, you will have to click and drag down the status bar to expand it which will give you detail as follows. This will be just 64 dp tall and called normal view.

Notification



4:23 PM
Thursday, October 2

The Big Meeting
4:15 – 5:15 PM
Big Conference Room

New Google+ notifications 3:44 PM
Earl Liibyrd: Added you back 2

Screenshot captured. 4:22 PM
Touch to view your screenshot.

Keep photos & videos backed.. 4:22 PM
Touch to get free private storage on Google

3 new messages 4:11 PM
kumlatar.swankatranami@gmail.com 3

3 new messages 2:42 PM
(754) 263-8267, 456

Create and Send Notifications-1

- Create Notification Builder
- As a first step is to create a notification builder using NotificationCompat.Builder.build().
- You will use Notification Builder to set various Notification properties like its small and large icons, title, priority etc.

```
NotificationCompat.Builder mBuilder = new
NotificationCompat.Builder(this)
```

Create and Send Notifications-2

- Setting Notification Properties
- Once you have Builder object, you can set its Notification properties using Builder object as per your requirement. But this is mandatory to set at least following –
  - A small icon, set by setSmallIcon()
  - A title, set by setContentTitle()
  - Detail text, set by setContentText()

```
mBuilder.setSmallIcon(R.drawable.notification_icon);
mBuilder.setContentTitle("Notification Alert, Click Me!");
mBuilder.setContentText("Hi, This is Android Notification
Detail!");
```

Create and Send Notifications-3

- Intent resultIntent = new Intent(this, ResultActivity.class);
- TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
- 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);
- mBuilder.setContentIntent(resultPendingIntent);

Create and Send Notifications-4

- Issue the notification:
- Finally, you pass the Notification object to the system by calling NotificationManager.notify() to send your notification. Make sure you call NotificationCompat.Builder.build() method on builder object before notifying it. This method combines all of the options that have been set and return a new Notification object.

```
NotificationManager mNotificationManager =
(NotificationManager) getSystemService
(Context.NOTIFICATION_SERVICE);

// notificationID allows you to update the notification
later on.

mNotificationManager.notify(notificationID,
mBuilder.build());
```

The NotificationCompat.Builder Class

- Notification build()
  - Combine all of the options that have been set and return a new Notification object.
- NotificationCompat.Builder setAutoCancel (boolean autoCancel)
  - Setting this flag will make it so the notification is automatically canceled when the user clicks it in the panel.
- NotificationCompat.Builder setContent (RemoteViews views)
  - Supply a custom RemoteViews to use instead of the standard one.
- NotificationCompat.Builder setContentInfo (CharSequence info)
  - Set the large text at the right-hand side of the notification.
- NotificationCompat.Builder setContentIntent (PendingIntent intent)
  - Supply a PendingIntent to send when the notification is clicked.

The NotificationCompat.Builder Class

- NotificationCompat.Builder setContentText (CharSequence text)
  - Set the text (second row) of the notification, in a standard notification.
- NotificationCompat.Builder setContentTitle (CharSequence title)
  - Set the text (first row) of the notification, in a standard notification.
- NotificationCompat.Builder setDefaults (int defaults)
  - Set the default notification options that will be used.
- NotificationCompat.Builder setLargeIcon (Bitmap icon)
  - Set the large icon that is shown in the ticker and notification.
- NotificationCompat.Builder setNumber (int number)
  - Set the large number at the right-hand side of the notification.

The NotificationCompat.Builder Class

- NotificationCompat.Builder setOngoing (boolean ongoing)
  - Set whether this is an ongoing notification.
- NotificationCompat.Builder setSmallIcon (int icon)
  - Set the small icon to use in the notification layouts.
- NotificationCompat.Builder setStyle (NotificationCompat.Style style)
  - Add a rich notification style to be applied at build time.
- NotificationCompat.Builder setTicker (CharSequence tickerText)
  - Set the text that is displayed in the status bar when the notification first arrives.
- NotificationCompat.Builder setVibrate (long[] pattern)
  - Set the vibration pattern to use.

Example:

- Notify.java

Thank you

Web Resources
http://mitu.co.in
http://tusharkute.com

Blogs
http://digitallocha.blogspot.in
http://kyamputar.blogspot.in

tushar@tusharkute.com