Mobile Computing

Android Notifications

Overview

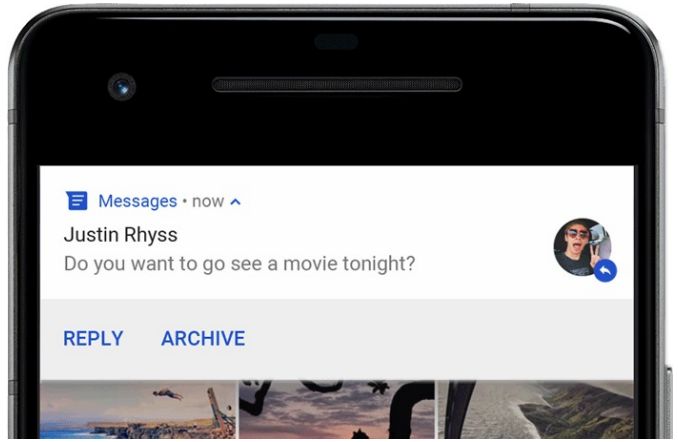
- A message that Android displays outside the app's UI
- provide the user with
  - Reminders
  - communication from other people
  - other timely information from the app
- Users can tap the notification to
  - open the app
  - take an action directly from the notification.

Appearances on a device

- Notifications appear to users in different locations and formats
  - as an icon in the status bar
    - first appears as an icon in the status bar
    - can drag down on a notification in the drawer to reveal more details
    - May contain action buttons
  - As a badge on the app's icon
  - on paired wearables

Heads-up notification

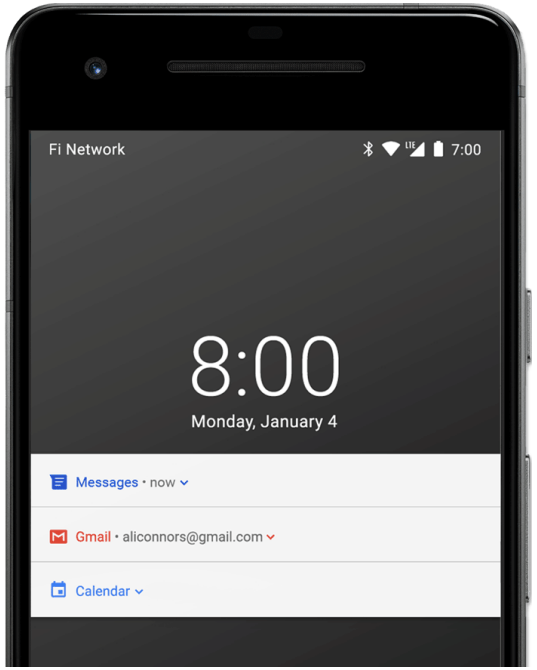
- notifications can briefly appear in a floating window
- for important notifications that the user should know about immediately
- appears only if the device is unlocked
- disappears after a moment
  - but remains visible in the notification drawer as usual
- Enabled with Android 5.0



Example conditions heads-up notifications

- The user's activity is in fullscreen mode
- The notification has high priority and uses ringtones or vibrations on devices running Android 7.1 (API level 25) and lower.
- The notification channel has high importance on devices running Android 8.0 (API level 26) and higher.

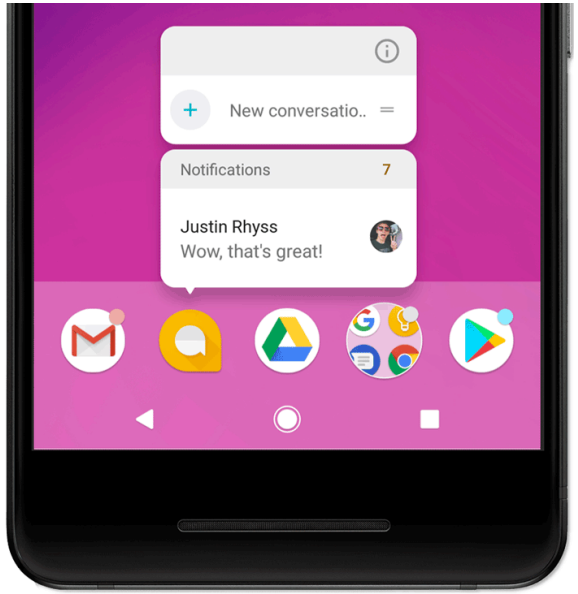
Lock screen



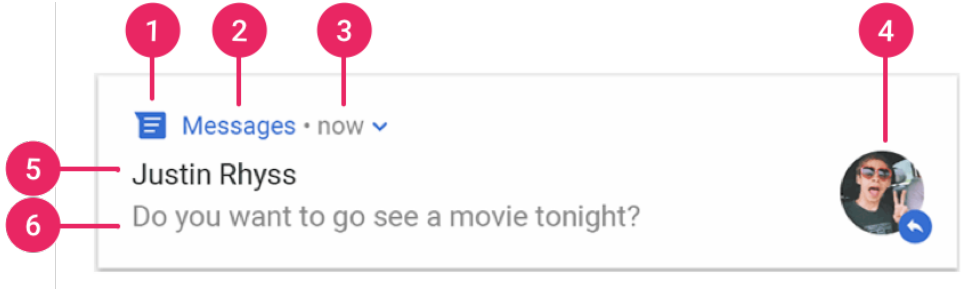
- Notifications can appear on the lock screen
- Support after android 5.0
- programmatically set the level of details
- Lock screen visibility
  - **VISIBILITY_PUBLIC** shows the notification's full content.
  - **VISIBILITY_SECRET** doesn't show any part of this notification on the lock screen.
  - **VISIBILITY_PRIVATE** shows basic information, such as the notification's icon and the content title, but hides the notification's full content.

App Icon Badge

- App icons indicate new notifications with a colored "badge" AKA "notification dot“
- long-press on an app icon shows the notifications for that app
  - Can dismiss or act on notifications from that menu, similar to the notification drawer.



Anatomy of notification



1. Small icon: This is required and set with **setSmallIcon**().
2. App name: This is provided by the system.
3. Time stamp: This is provided by the system but you can override with **setWhen**() or hide it with **setShowWhen**(**false**).
4. Large icon: This is optional (usually used only for contact photos; do not use it for your app icon) and set with **setLargeIcon**().
5. Title: This is optional and set with **setContentTitle**().
6. Text: This is optional and set with **setContentText**().

Remove a notification

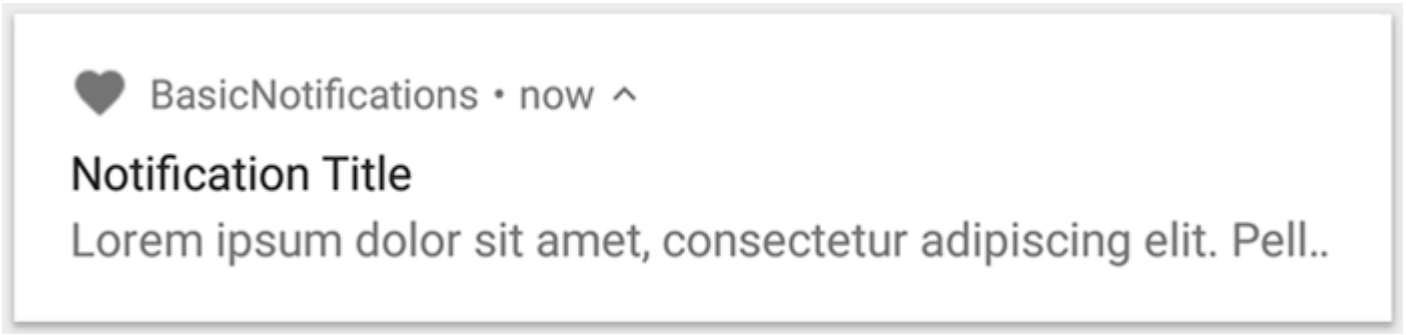
- Notifications remain visible until one of the following happens:
  - The user dismisses the notification.
  - The user clicks the notification, and you called **setAutoCancel**() when you created the notification.
  - You call **cancel**() for a specific notification ID. This method also deletes ongoing notifications.
  - You call **cancelAll**(), which removes all of the notifications you previously issued.
  - set a timeout when creating a notification using **setTimeoutAfter**(), the system cancels the notification after the specified duration elapses.

Good Practice

- Decide the priority of the notifications accurately
- Update the existing notification as possible as you can
- Do not create multiple notifications in quick successions
- Use action buttons in notification to direct the user to correct context of the application
- If possible, use the inbuilt text box to complete the task related to the notification.
- Keep in mind, that foreground services always show notifications that cannot be removed.

Create a Notification

- Basic notification:




```
NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
.setSmallIcon(R.drawable.notification_icon)
.setContentTitle(textTitle) .setContentText(textContent)
.setPriority(NotificationCompat.PRIORITY_DEFAULT);
```

Create a Notification Cont.

- Expandable Notification

```
NotificationCompat.Builder builder =
new NotificationCompat.Builder(this, CHANNEL_ID)
.setSmallIcon(R.drawable.notification_icon)
.setContentTitle("My notification")
.setContentText("Much longer text that cannot fit one
line...")
.setStyle(new NotificationCompat.BigTextStyle()
      .bigText("Much longer text that cannot fit one
line..."))
.setPriority(NotificationCompat.PRIORITY_DEFAULT);
```

Create a channel

- Android 8.0 and higher, you must register your app's notification channel

```
private void createNotificationChannel() {
 if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) { CharSequence name =
      getString(R.string.channel_name); String description =
      getString(R.string.channel_description); int importance =
      NotificationManager.IMPORTANCE_DEFAULT; NotificationChannel channel =
            new NotificationChannel(CHANNEL_ID, name, importance);
      channel.setDescription(description);
      NotificationManager notificationManager =
      getSystemService(NotificationManager.class);
      notificationManager.createNotificationChannel(channel);
      }
}
```

Set the notification's tap action

```
Intent intent = new Intent(this, AlertDetails.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
PendingIntent pendingIntent = PendingIntent.getActivity(this,
0, intent, 0);

NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
.setSmallIcon(R.drawable.notification_icon)
.setContentTitle("My notification")
.setContentText("Hello World!")
.setPriority(NotificationCompat.PRIORITY_DEFAULT)
.setContentIntent(pendingIntent)
.setAutoCancel(true);
```

Show the notification

```
NotificationManagerCompat notificationManager
= NotificationManagerCompat.from(this);
notificationManager.notify(notificationId,
builder.build());
```