

# Notifications Overview

A notification is a message that Android displays outside your app's UI to provide the user with reminders, communication from other people, or other timely information from your app. Users can tap the notification to open your app or take an action directly from the notification.

This page provides an overview of where notifications appear and the available features. If you want to start building your notifications, instead read [Create a Notification](#).

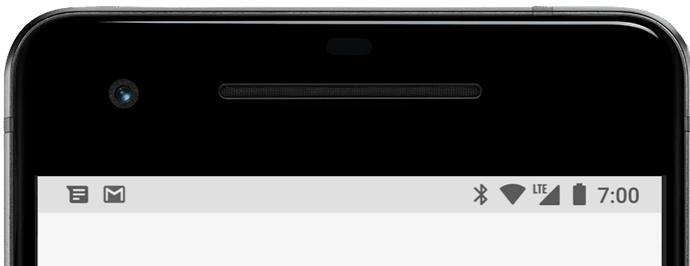
For more information about the design and interaction patterns, see the [Notifications design guide](#). Additionally, see the [Android Notifications Sample](#) for a demonstration of best practices in using the `Notification.Style` API in both mobile and wearable apps.

## Appearances on a device

Notifications appear to users in different locations and formats, such as an icon in the status bar, a more detailed entry in the notification drawer, as a badge on the app's icon, and on paired wearables automatically.

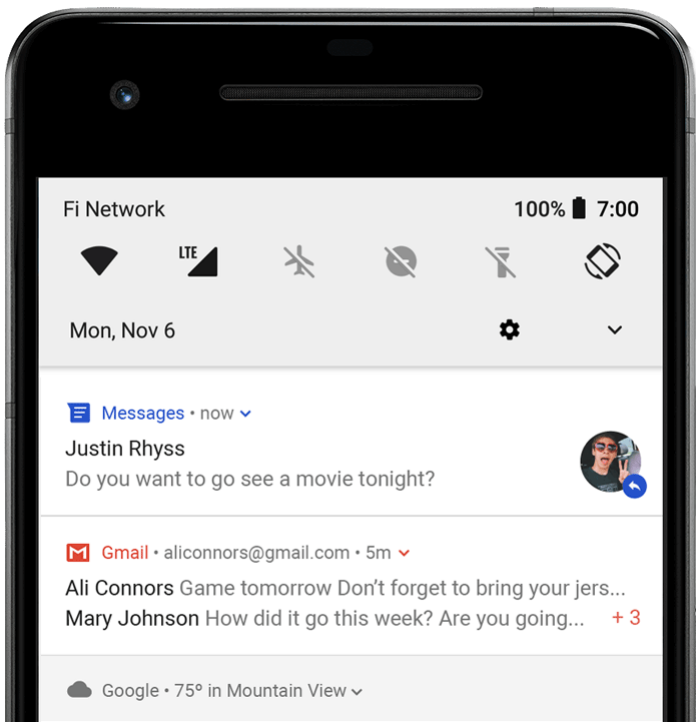
### Status bar and notification drawer

When you issue a notification, it first appears as an icon in the status bar.



**Figure 1.** Notification icons appear on the left side of the status bar

Users can swipe down on the status bar to open the notification drawer, where they can view more details and take actions with the notification.



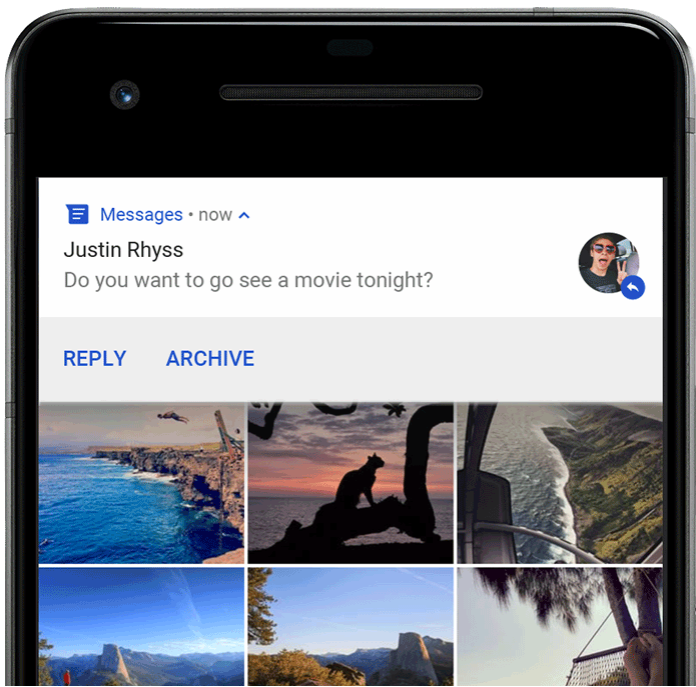
**Figure 2.** Notifications in the notification drawer

Users can drag down on a notification in the drawer to reveal the expanded view, which shows additional content and action buttons, if provided.

A notification remains visible in the notification drawer until dismissed by the app or the user.

## Heads-up notification

Beginning with Android 5.0, notifications can briefly appear in a floating window called a *heads-up notification*. This behavior is normally for important notifications that the user should know about immediately, and it appears only if the device is unlocked.



**Figure 3.** A heads-up notification appears in front of the foreground app

The heads-up notification appears the moment your app issues the notification and it disappears after a moment, but remains visible in the notification drawer as usual.

Example conditions that might trigger heads-up notifications include the following:

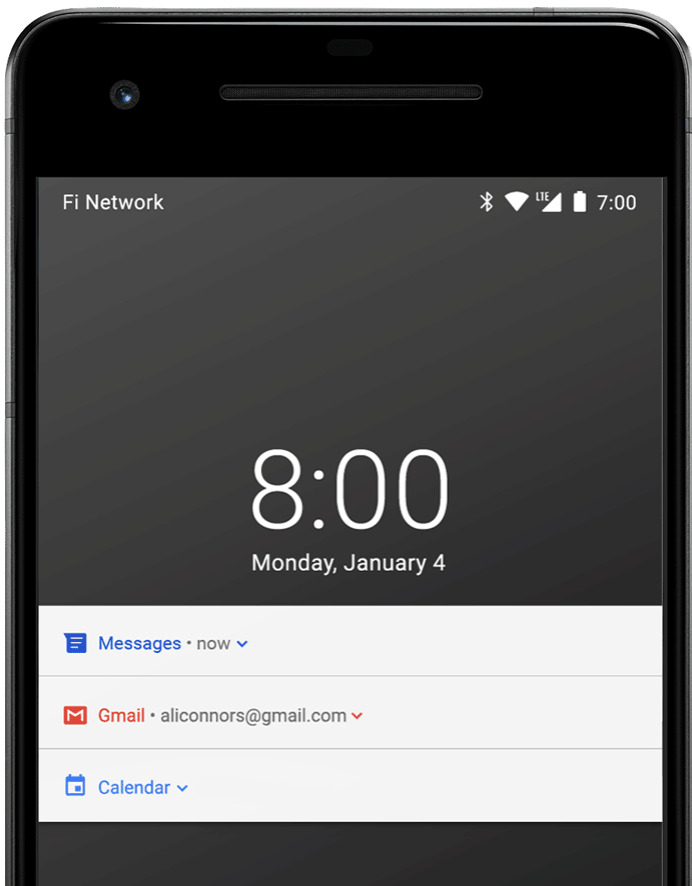
- The user's activity is in fullscreen mode (the app uses [fullScreenIntent](#)).
- The notification has high priority and uses ringtones or vibrations on devices running Android 7.1 (API level 25) and lower.
- The notification channel has high importance on devices running Android 8.0 (API level 26)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

## Lock screen

Beginning with Android 5.0, notifications can appear on the lock screen.

You can programmatically set the level of detail visible in notifications posted by your app on a secure lock screen, or even whether the notification will show on the lock screen at all.

Users can use the system settings to choose the level of detail visible in lock screen notifications, including the option to disable all lock screen notifications. Starting with Android 8.0, users can choose to disable or enable lock screen notifications for each [notification channel](#).



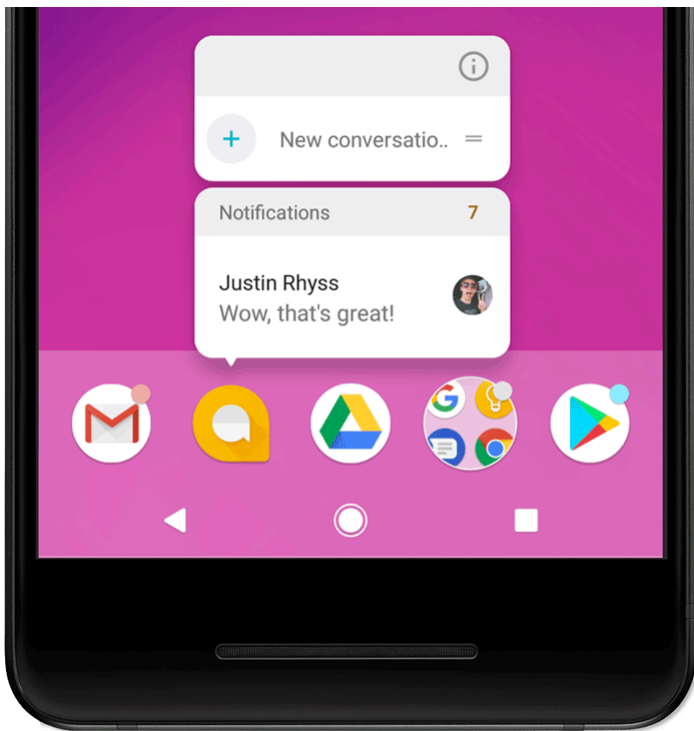
**Figure 4.** Notifications on the lock screen with sensitive content hidden

To learn more, see how to [Set lock screen visibility](#).

## App icon badge

In supported launchers on devices running Android 8.0 (API level 26) and higher, app icons indicate new notifications with a colored "badge" (also known as a "notification dot") on the corresponding app launcher icon.

Users can long-press on an app icon to see the notifications for that app. Users can then dismiss or act on notifications from that menu, similar to the notification drawer.



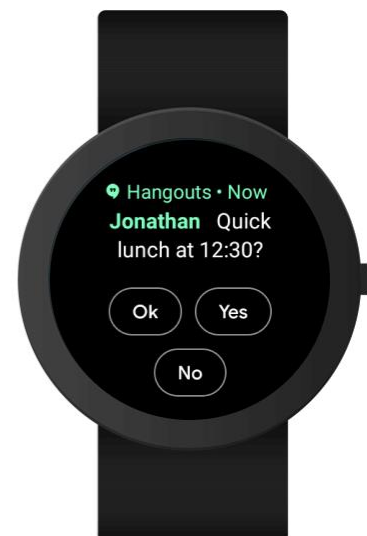
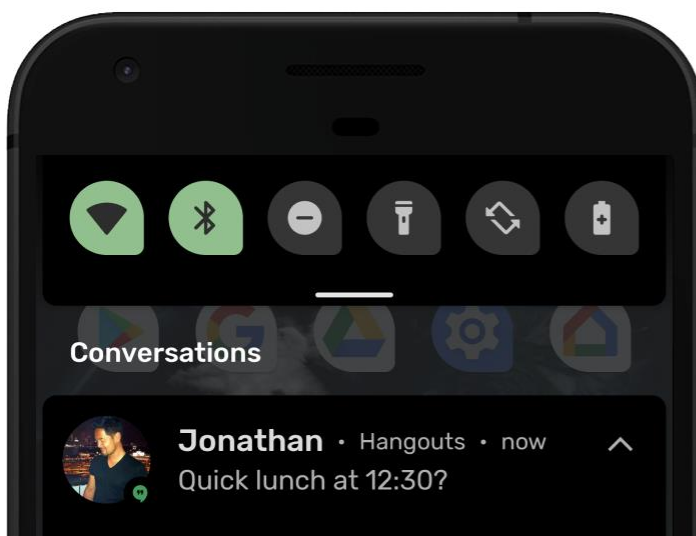
**Figure 5.** Notification badges and the long-press menu

To learn more about how badges work, read [Notification badges](#).

## Wear OS devices

If the user has a paired Wear OS device, all your notifications appear there automatically, including expandable detail and action buttons.

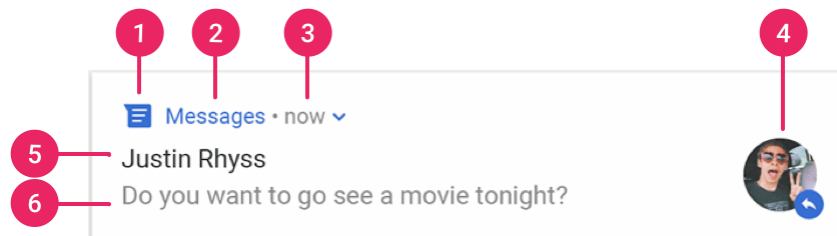
You can also enhance the experience by customizing some appearances for the notification on wearables and provide different actions, including suggested replies and voice input replies. For more information, see how to [add wearable-specific features to your notification](#).



**Figure 6.** Notifications automatically appear on a paired Wear OS device

## Notification anatomy

The design of a notification is determined by system templates—your app simply defines the contents for each portion of the template. Some details of the notification appear only in the expanded view.



**Figure 7.** A notification with basic details

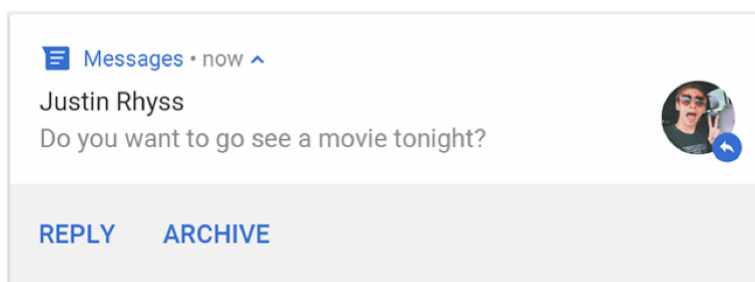
The most common parts of a notification are indicated in figure 7 as follows:

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\(\)](#).

For more information about how to create a notification with these features and more, read [Create a Notification](#).

## Notification actions

Although it's not required, every notification should open an appropriate app activity when tapped. In addition to this default notification action, you can add action buttons that complete an app-related task from the notification (often without opening an activity), as shown in figure 9.



**Figure 9.** A notification with action buttons

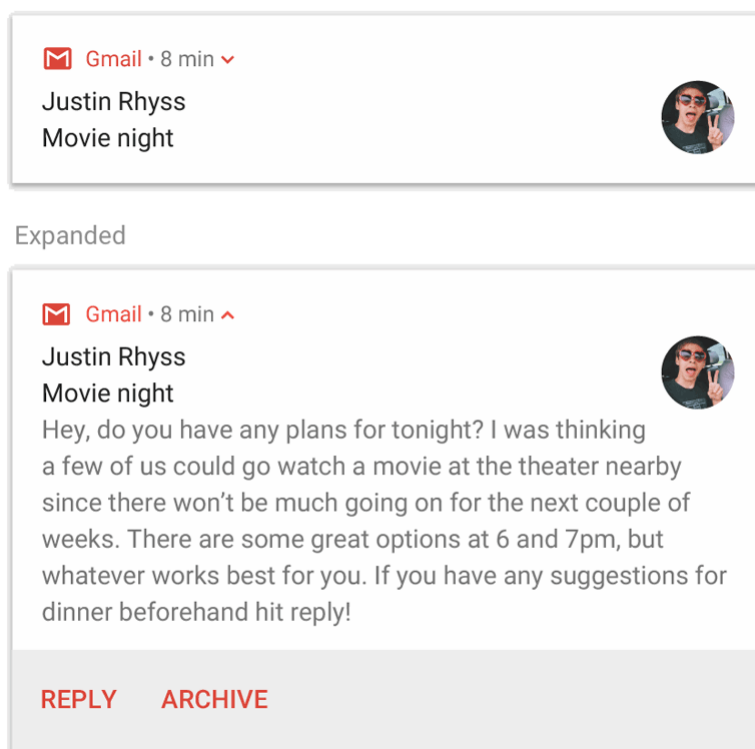
Starting in Android 7.0 (API level 24), you can also add an action to reply to messages or enter other text directly from the notification.

Starting in Android 10 (API level 29), the platform can automatically generate action buttons with suggested intent-based actions.

Adding action buttons is explained further in [Create a Notification](#).

## Expandable notification

By default, the notification's text content is truncated to fit on one line. If you want your notification to be longer, you can enable a larger text area that's expandable by applying an additional template, as shown in figure 8.



**Figure 8.** An expandable notification for large text

You can also create an expandable notification with an image, in inbox style, a chat conversation, or media playback controls. For more information, read [Create an Expandable Notification](#).

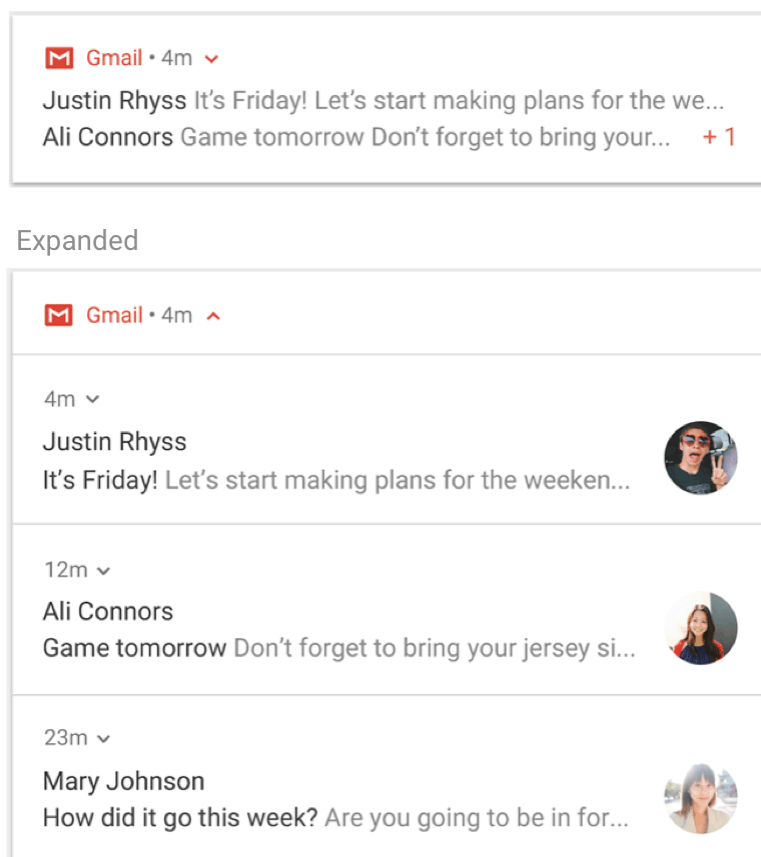
And although we recommend you always use these templates to ensure proper design compatibility on all devices, if necessary, you can also [create a custom notification layout](#).

## Notification updates and groups

To avoid bombarding your users with multiple or redundant notifications when you have additional updates, you should consider [updating an existing notification](#) rather than issuing a new one, or consider using the [inbox-style notification](#) to show conversation updates.

However, if it's necessary to deliver multiple notifications, you should consider grouping those separate notifications into a group (available on Android 7.0 and higher). A notification group allows you to collapse multiple notifications into just one post in the notification drawer, with a summary. The user can then expand the notification to reveal the details for each individual notification.

The user can progressively expand the notification group and each notification within it for more details.



**Figure 10.** A collapsed and expanded notification group

To learn how to add notifications to a group, see [Create a Group of Notifications](#).

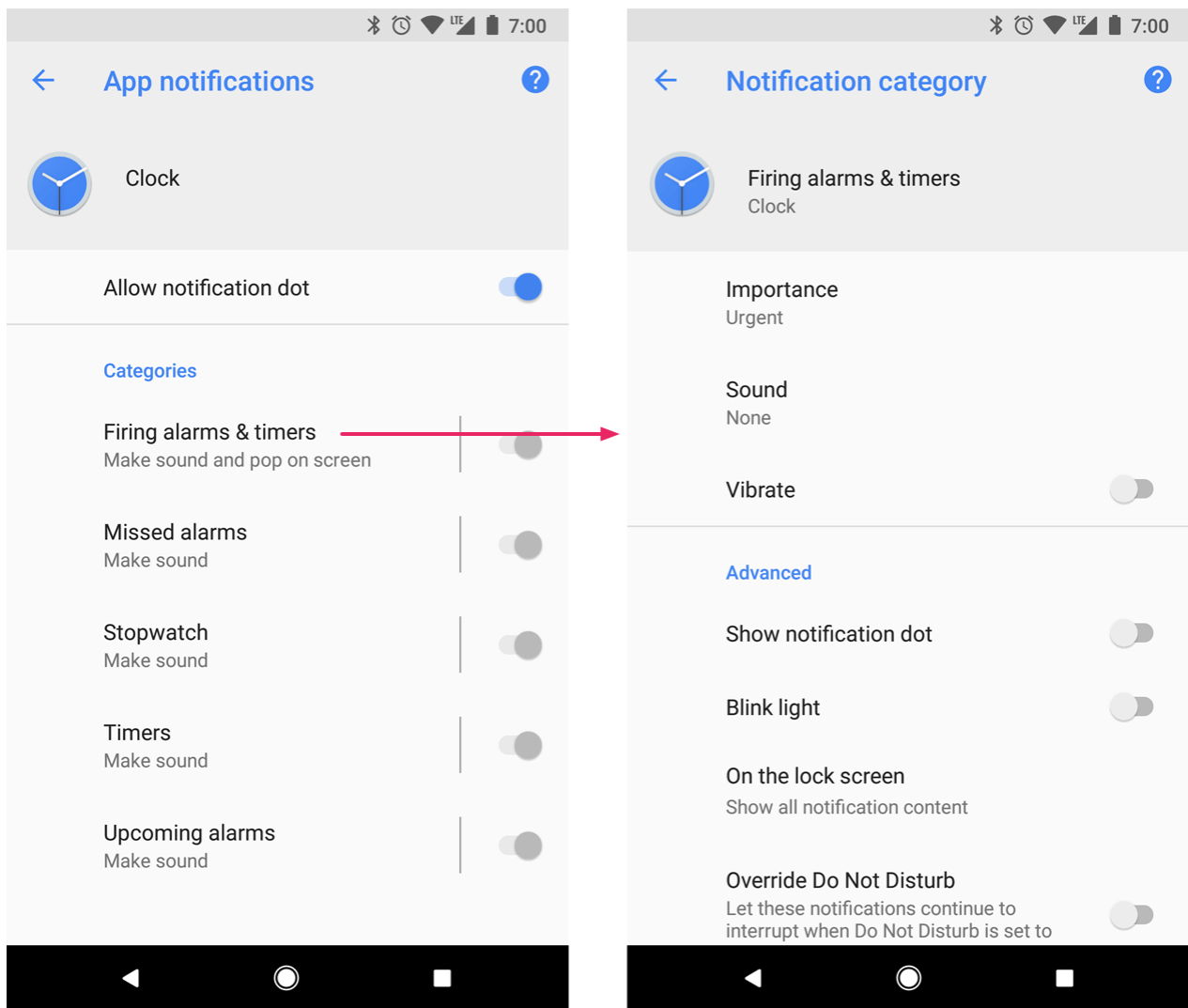
**Note:** If the same app sends four or more notifications and does not specify a grouping, the system automatically groups them together.

## Notification channels

Starting in Android 8.0 (API level 26), all notifications must be assigned to a channel or it will not appear. By categorizing notifications into channels, users can disable specific notification channels for your app (instead of disabling *all* your notifications), and users can control the visual and auditory options for each channel—all from the Android system settings (figure 11). Users can also long-press a notification to change behaviors for the associated channel.

On devices running Android 7.1 (API level 25) and lower, users can manage notifications on a per-app basis only (effectively each app only has one channel on Android 7.1 and lower).





**Figure 11.** Notification settings for Clock app and one of its channels

**Note:** The user interface refers to channels as "categories."

One app can have multiple notification channels—a separate channel for each type of notification the app issues. An app can also create notification channels in response to choices made by users of your app. For example, you may set up separate notification channels for each conversation group created by a user in a messaging app.

The channel is also where you specify the [importance level](#) for your notifications on Android 8.0 and higher. So all notifications posted to the same notification channel have the same behavior.

For more information, see [Create and Manage Notification Channels](#).

## Notification importance

Android uses the *importance* of a notification to determine how much the notification should interrupt the user (visually and audibly). The higher the importance of a notification, the more interruptive the notification will be.

On Android 8.0 (API level 26) and above, importance of a notification is determined by the [importance](#) of the channel the notification was posted to. Users can change the importance of a notification channel in the system settings (figure 12). On Android 7.1 (API level 25) and below, importance of each notification is determined by the notification's [priority](#).



## Importance



**Urgent**

Make sound and pop on screen



**High**

Make sound



**Medium**

No sound



**Low**

No sound or visual interruption



**Figure 12.** Users can change the importance of each channel on Android 8.0 and higher

The possible importance levels are the following:

- Urgent: Makes a sound and appears as a heads-up notification.
- High: Makes a sound.
- Medium: No sound.
- Low: No sound and does not appear in the status bar.

All notifications, regardless of importance, appear in non-interruptive system UI locations, such as in the notification drawer and as a badge on the launcher icon (though you can [modify the appearance of the notification badge](#)).

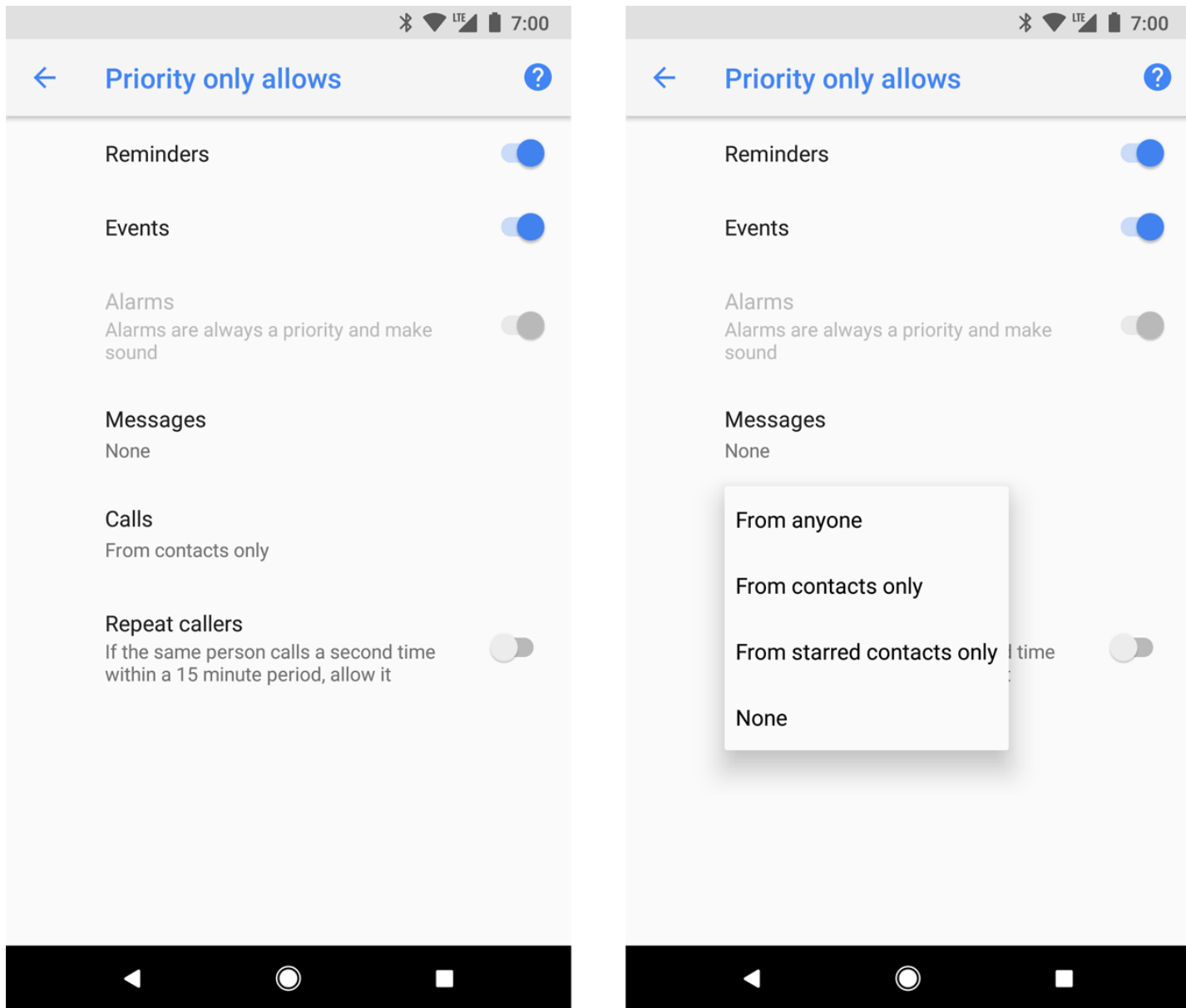
For more information, see [how to set the importance](#).

## Do Not Disturb mode

Starting in Android 5.0 (API level 21), users can enable *Do Not Disturb mode*, which silences sounds and vibration for all notifications. Notifications still appear in the system UI as normal, unless the user specifies otherwise.

There are three different levels available in Do Not Disturb mode:

- Total silence: blocks all sounds and vibrations, including from alarms, music, videos, and games.
- Alarms only: blocks all sounds and vibrations, except from alarms.
- Priority only: users can configure which system-wide categories can interrupt them (such as only alarms, reminders, events, calls, or messages). For messages and calls, users can also choose to filter based on who the sender or caller is (figure 13).



**Figure 13.** Users can allow notifications through based on system-wide categories (left) and based on who sent a message or who is calling (right).

On Android 8.0 (API level 26) and above, users can additionally allow notifications through for app-specific categories (also known as channels) by overriding Do Not Disturb on a channel-by-channel basis. For example, a payment app might have channels for notifications related to withdrawals and deposits. The user can then choose to allow either withdrawal notifications, deposit notifications, or both when in priority mode. On devices running Android 7.1 (API level 25) and below, users can allow notifications through on an app by app basis, rather than on a channel by channel basis.

To properly configure your notifications for these user settings, you must [set a system-wide category and channel](#).

## Notifications for foreground services

A notification is required when your app is running a "foreground service"—a [Service](#) running in the background that's long living and noticeable to the user, such as a media player. This notification cannot be dismissed like other notifications. To remove the notification, the service must be either stopped or removed from the "foreground" state.

For more information read [Running a service in the foreground](#). And if you are building a media player, also read [Using MediaStyle notifications with a foreground service](#).

## Posting limits

Beginning with Android 8.1 (API level 27), apps cannot make a notification sound more than once per second. If your app posts multiple notifications in one second, they all appear as expected, but only the first notification per second makes a sound.

However, Android also applies a rate limit when updating a notification. If you post updates to a single notification too frequently (many in less than one second), the system might drop some updates.

## Notification compatibility

Since Android 1.0, the notification system UI and the notification-related APIs have continually evolved. To use the latest notification API features while still supporting older devices, use the support library notification API: [NotificationCompat](#) and its subclasses, as well as [NotificationManagerCompat](#). This will allow you to avoid writing conditional code to check API levels because these APIs handle that for you.

[NotificationCompat](#) is updated as the platform evolves to include the latest methods. It is important to note that the availability of a method in [NotificationCompat](#) does not guarantee that the corresponding feature will be provided on older devices. In some cases calling a newly-introduced API results in a no-op on older devices. For example, [NotificationCompat.addAction\(\)](#) displays the action button on a device running Android 4.1 (API level 16) and higher only.

The following is a summary of the most notable behavior changes for Android notifications.

### Android 4.1, API level 16

- Introduced expandable notification templates (called notification styles), allowing for larger notification content area to display information. Users can use a one finger swipe up/down gesture to expand a notification.
- Also introduced the ability to add additional actions, in the form of buttons, to a notification.
- Added ability for users to turn notifications off on a per-app basis in settings.

### Android 4.4, API level 19 and 20

- Notification listener services were added to the API.
- Android Wear (now called Wear OS) support was added in API level 20.

### Android 5.0, API level 21

- Introduced lock screen and heads-up notifications.
- The user can now put the phone into Do Not Disturb mode and configure which notifications are allowed to interrupt them when the device is in priority only mode.
- Methods added to API set whether or not a notification is displayed on the lock screen ([setVisibility\(\)](#)) and for specifying “public” version of the notification text.
- [setPriority\(\)](#) method added which tells the system how “interruptive” this notification should be (e.g. setting it to high makes the notification appear as a heads-up notification).
- Notification stacks support added to Android Wear (now called Wear OS) devices. Put notifications into a stack using [setGroup\(\)](#). Note that notification stacks were not supported on tablets nor phones yet. Notification stacks would later become known as a group or bundle.

### **Android 7.0, API level 24**

- Notification templates were restyled to put emphasis on the hero image and avatar.
- Three notification templates were added: one for messaging apps and the other two for decorating custom content views with the expandable affordance and other system decorations.
- Support added to handheld devices (phones and tablets) for notification groups. Uses the same API as Android Wear (now called Wear OS) notification stacks introduced in Android 5.0 (API level 21).
- Users can reply directly inside of a notification (they can enter text which will then be routed to the notification’s parent app) using inline reply.

### **Android 8.0, API level 26**

- Individual notifications must now be put in a specific channel.
- Users can now turn off notifications per [channel](#), instead of turning off all notifications from an app.
- Apps with active notifications display a notification "badge" on top of their app icon on the home/launcher screen.
- Users can now snooze a notification from the drawer. You can set an automatic timeout for a notification.
- You can also set the notification's background color.
- Some APIs regarding notification behaviors were moved from [Notification](#) to [NotificationChannel](#). For example, use [NotificationChannel.setImportance\(\)](#) instead of [NotificationCompat.Builder.setPriority\(\)](#) for Android 8.0 and higher.

[Next](#)  
[arrow\\_forward Create a Notification](#)

<https://developer.android.com/guide/topics/ui/notifiers/notifications?hl=vi>