Android Development

MNNIT Computer Coding Club


Tapped on phone 7 times
NOW A DEVELOPER

Objective

To make you answer following questions yourselves:

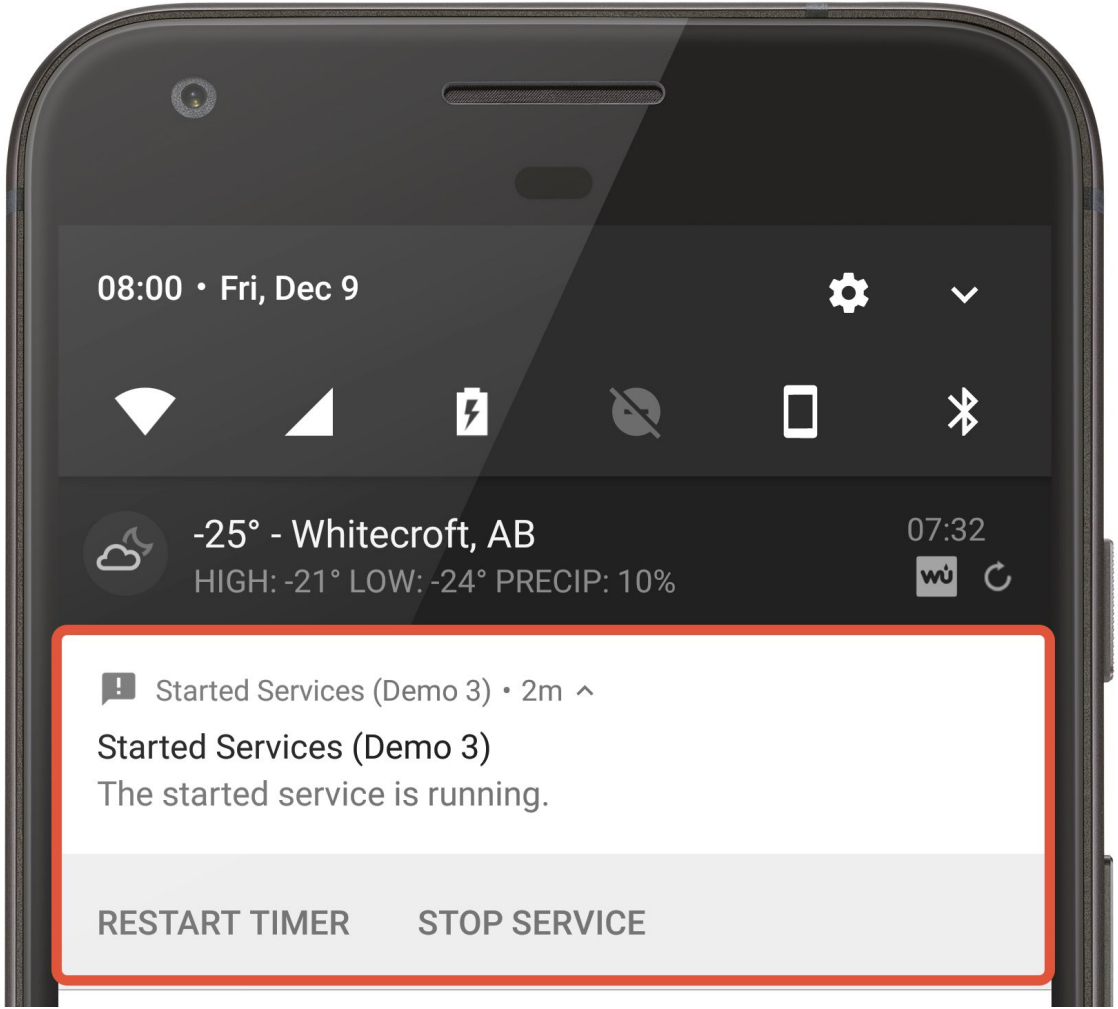
1. What are Services?
2. How to work with JSON?
3. How to make database?
4. What are notifications?

Services

A Service is an application component that can perform long-running operations in the background. It does not provide a user interface. Once started, a service might continue running for some time, even after the user switches to another application. Additionally, a component can bind to a service to interact with it and even perform interprocess communication (IPC). For example, a service can handle network transactions, play music, perform file I/O, or interact with a content provider, all from the background. There are three types of Services: Foreground Service, Background Service and Bound Service.

Foreground Service

A foreground service performs some operation that is noticeable to the user. For example, an audio app would use a foreground service to play an audio track. Foreground services must display a Notification. Foreground services continue running even when the user isn't interacting with the app.When you use a foreground service, you must display a notification so that users are actively aware that the service is running. This notification cannot be dismissed unless the service is either stopped or removed from the foreground.


08:00 • Fri, Dec 9
-25° - Whitecroft, AB
HIGH: -21° LOW: -24° PRECIP: 10%
07:32
Started Services (Demo 3) • 2m
Started Services (Demo 3)
The started service is running.
RESTART TIMER
STOP SERVICE

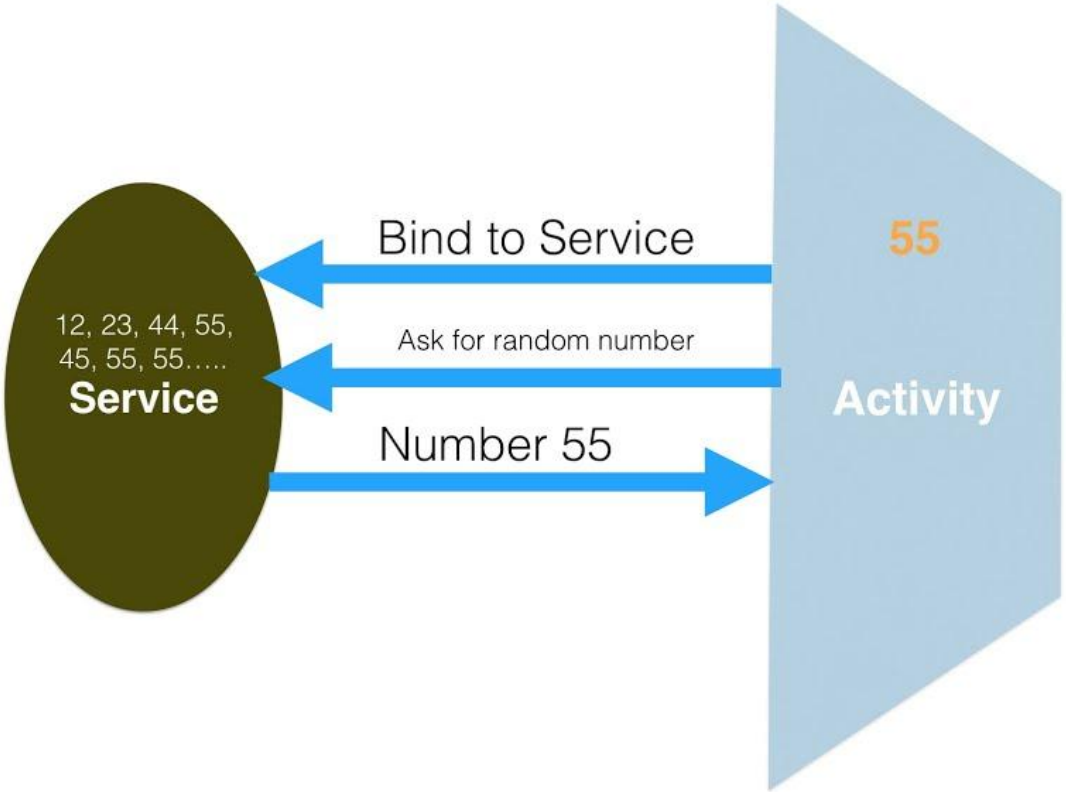
Background Service

A background service performs an operation that isn't directly noticed by the user. For example, if an app used a service to compact its storage, that would usually be a background service.

Bound Service

A service is *bound* when an application component binds to it by calling bindService(). A bound service offers a client-server interface that allows components to interact with the service, send requests, receive results, and even do so across processes with interprocess communication (IPC). A bound service runs only as long as another application component is bound to it. Multiple components can bind to the service at once, but when all of them unbind, the service is destroyed.

Local binding preparation (IBinder)



Choosing between a service and a thread

A service is simply a component that can run in the background, even when the user is not interacting with your application, so you should create a service only if that is what you need.

If you must perform work outside of your main thread, but only while the user is interacting with your application, you should instead create a new thread in the context of another application component. For example, if you want to play some music, but only while your activity is running, you might create a thread in onCreate(), start running it in onStart(), and stop it in onStop(). Also consider using thread pools and executors from the java.util.concurrent.

Important

A service runs in the main thread of its hosting process; the service does not create its own thread and does not run in a separate process unless you specify otherwise. You should run any blocking operations on a separate thread within the service to avoid Application Not Responding (ANR) errors.

JSON

JSON stands for JavaScript Object Notation. JSON is a lightweight format for storing and transporting data. JSON is often used when data is sent from a server to a web page.

It is easy for humans to read and write. It is easy for machines to parse and generate.

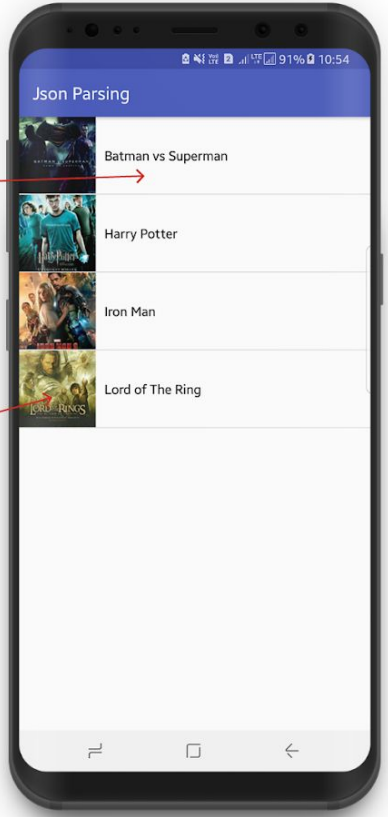
Example: https://api.github.com/users/ashu12chi

JSON parsing

JSONObject
{ - represents its a json Object node

JSONArray
[- represents its a json Array node

```
{
    "movies": [
        {
            "name": "Batman vs Superman",
            "imageurl": "http://www.snowcorp.org/assets/demo/batman-v-superman.jpg"
        },
        {
            "name": "Harry Potter",
            "imageurl": "http://www.snowcorp.org/assets/demo/harry-potter.jpg"
        },
        {
            "name": "Iron Man",
            "imageurl": "http://www.snowcorp.org/assets/demo/iron-man.jpg"
        },
        {
            "name": "Lord of The Ring",
            "imageurl": "http://www.snowcorp.org/assets/demo/lotr.jpg"
        }
    ]
}
```



JSON Parsing Example

Let’s see an example of JSON Parsing using github api. In this app, we will fetch user details by sending request to: https://api.github.com/users/ashu12chi

Firebase

Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app development.

Now, let’s discuss what firebase offers us.

Firebase

Build better apps

- Auth
- Hosting
- Cloud Functions
- ML Kit
- Cloud Firestore
- Realtime Database
- Cloud Storage

Improve app quality

- Crashlytics
- Performance Monitoring
- Test Lab

Grow your app

- Analytics
- Remote Config
- Predictions
- A/B Testing
- Cloud Messaging
- Dynamic Links
- In-app Messaging

Build

Build powerful apps. Spin up your backend without managing servers. Effortlessly scale to support millions of users with Firebase databases, machine learning infrastructure, hosting and storage solutions, and Cloud Functions.

As a beginner you should start considering Firebase Authentication, Realtime database, Cloud Firestore, Cloud Storage and may be later you can consider Cloud Functions too.

Release and Monitor

Improve app quality in less time with less effort. Simplify testing, triaging, and troubleshooting. Carefully roll out features and monitor adoption. Pinpoint, prioritize, and fix stability and performance issues early.

As a beginner you can ignore these things.

Engage

Boost user engagement with rich analytics, A/B testing, and messaging campaigns. Understand your users to better support and retain them. Run experiments to test ideas and uncover new insights. Customize your app for different user segments.

As a beginner you can consider learning Cloud Messaging.

Home Assignment

Study about history of firebase, trust us it’s interesting.

Firebase Project Example

Let’s now see one of our project: BrighterBee in firebase console.

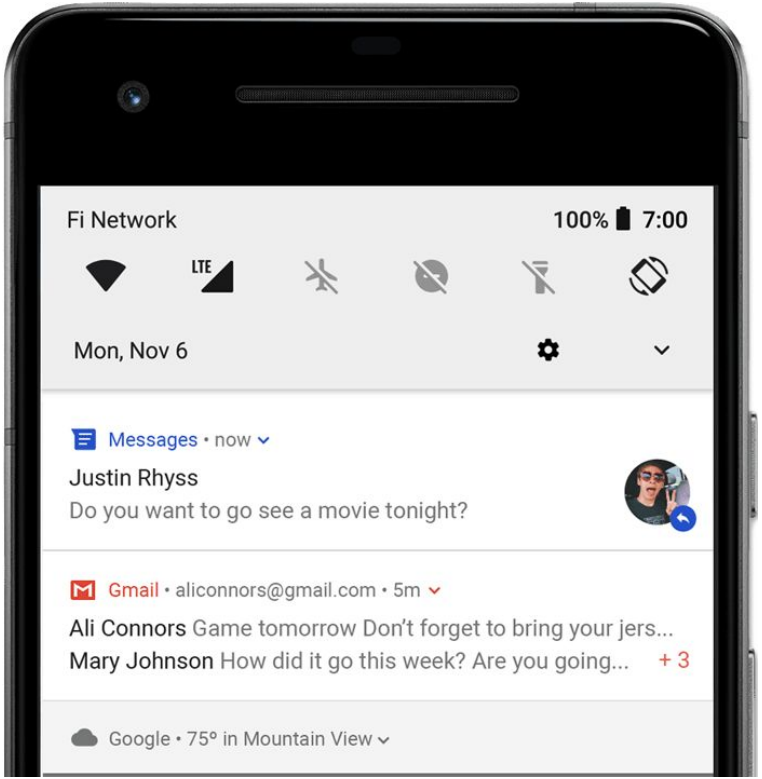
Notifications

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.

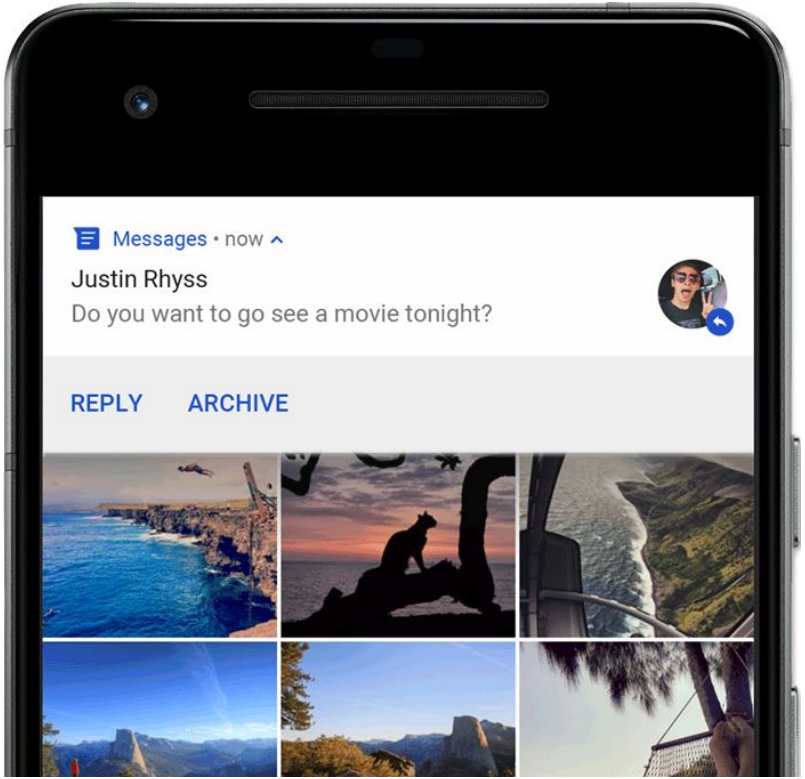
Notifications appear to users in different locations and formats, such as an icon in the status bar, a more detailed entry in the notification drawer, as a badge on the app's icon, and on paired wearables automatically.

Status bar

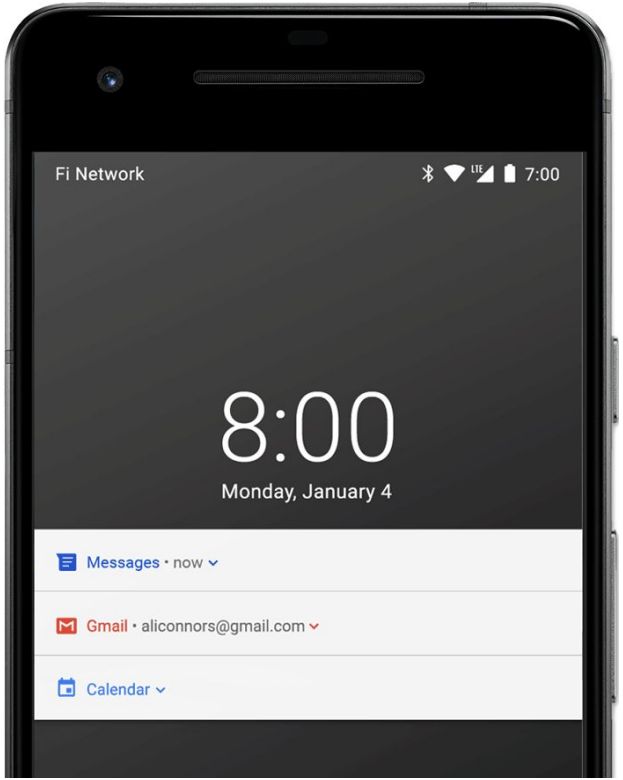
Notification Drawer



Heads-up Notification



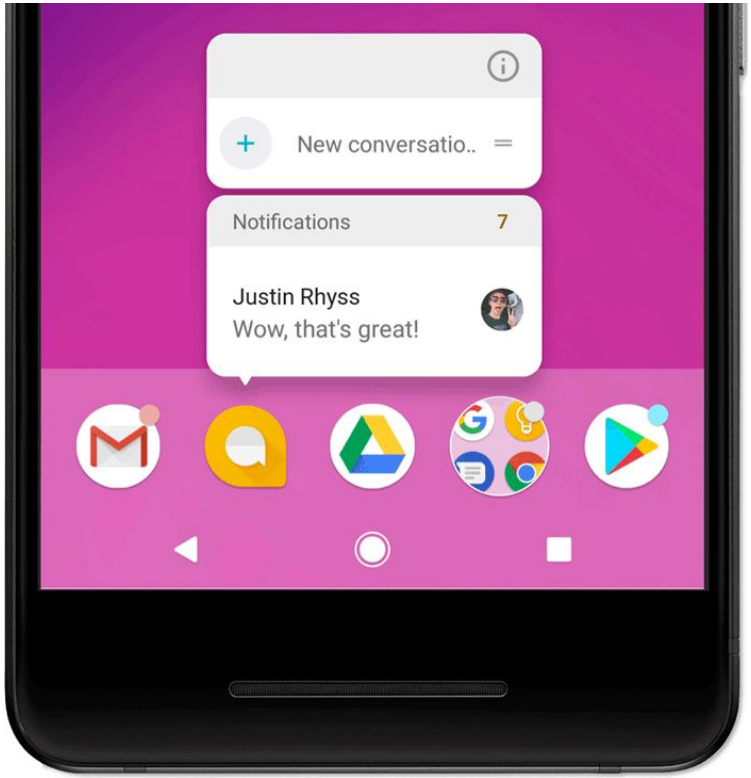
Lock Screen


Fi Network
7:00
8:00
Monday, January 4
Messages • now
Gmail • aliconnors@gmail.com
Calendar

Home Assignment

From which version of android, notifications start appearing on lock screen.

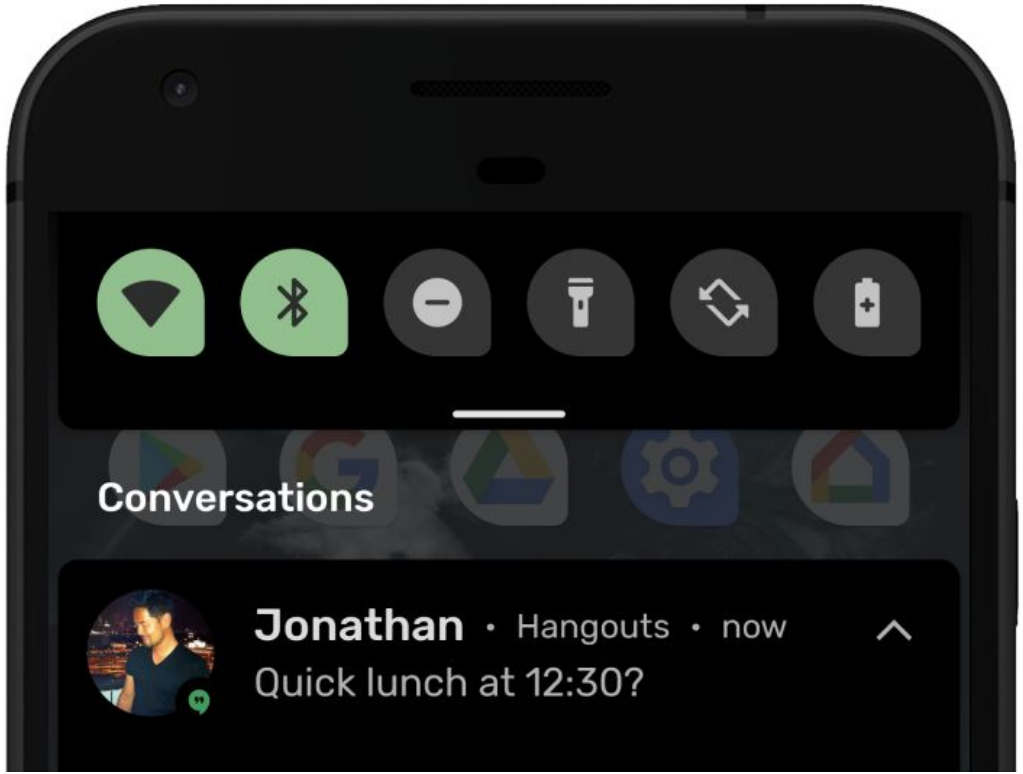
App Icon Badge

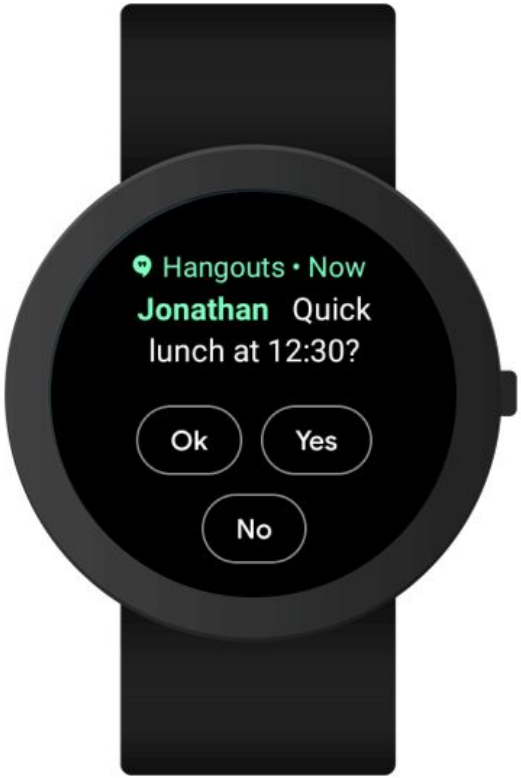


Home Assignment

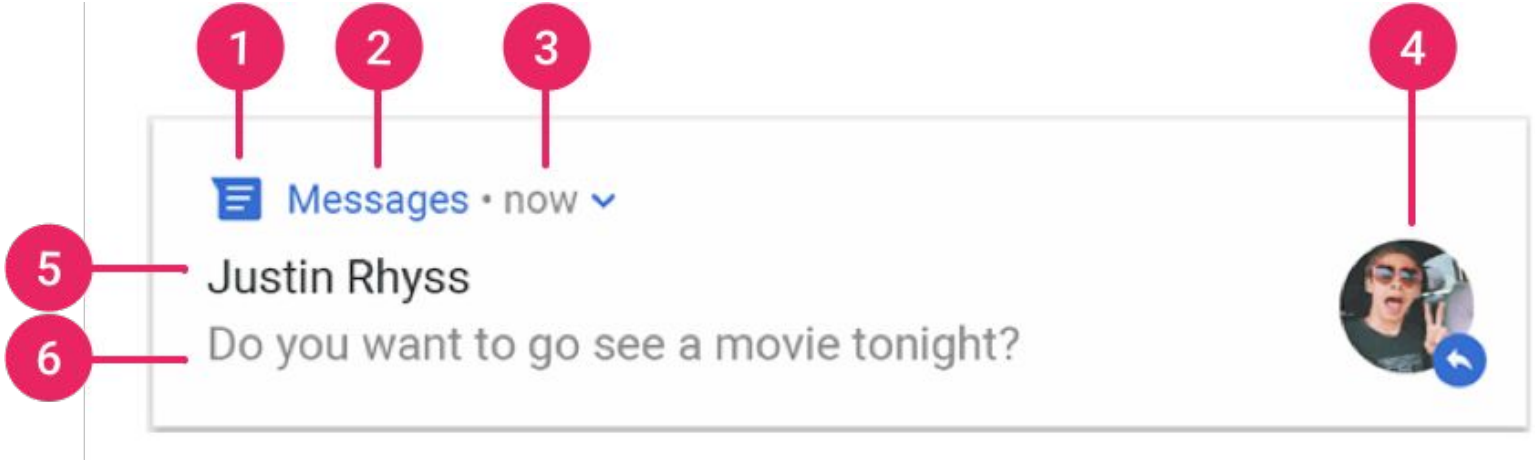
From which version of android, notifications start appearing on App icons.

Wear OS Devices





Notification Anatomy



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()`.

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).

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.

Messages • now

Justin Rhyss

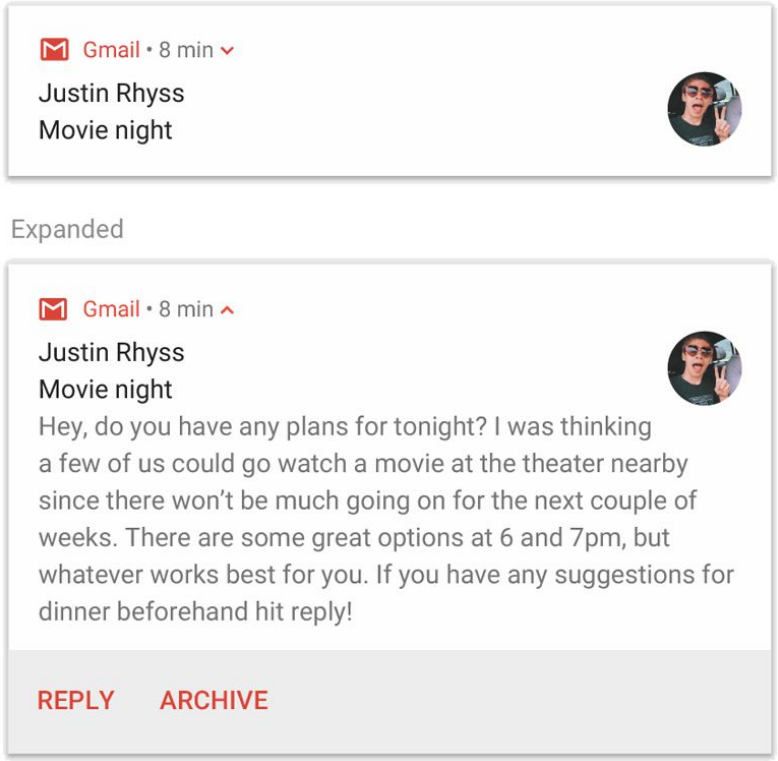
Do you want to go see a movie tonight?

REPLY ARCHIVE

Expandable Notifications

By default, the notification 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.

You can also create an expandable notification with an image, in inbox style, a chat conversation, or media playback controls.



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.

Gmail • 4m

Justin Rhyss It's Friday! Let's start making plans for the we...
Ali Connors Game tomorrow Don't forget to bring your... + 1

Expanded

Gmail • 4m

4m

Justin Rhyss
It's Friday! Let's start making plans for the weeken...

12m

Ali Connors
Game tomorrow Don't forget to bring your jersey si...

23m

Mary Johnson
How did it go this week? Are you going to be in for...

User Location

To protect user privacy, apps that use location services must request location permissions.

When you request location permissions, follow the same best practices as you would for any other runtime permission. One important difference when it comes to location permissions is that the system includes multiple permissions related to location. Which permissions you request, and how you request them, depend on the location requirements for your app's use case. There are two types of location access: Foreground location and Background location.

Foreground Location

If your app contains a feature that shares or receives location information only once, or for a defined amount of time, then that feature requires foreground location access. One example is within a messaging app, a feature allows users to share their current location with another user.

The system considers your app to be using foreground location if a feature of your app accesses the device's current location in one of the following situations:

1. An activity that belongs to your app is visible.
2. Your app is running a foreground service. When a foreground service is running, the system raises user awareness by showing a persistent notification.

Foreground Location Continued...

You declare a need for foreground location when your app requests either the ACCESS_COARSE_LOCATION permission or the ACCESS_FINE_LOCATION permission, as shown in the following snippet:

```
<manifest ... >
  <!-- To request foreground location access, declare one of these permissions. -->
  <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
</manifest>
```

Foreground Location Continued...

The level of precision depends on which permission you request:

1. ACCESS_COARSE_LOCATION: Provides location accuracy to within a city block.
2. ACCESS_FINE_LOCATION: Provides a more accurate location than one provided when you request ACCESS_COARSE_LOCATION. This permission is necessary for some connectivity tasks.

Background Location

An app requires background location access if a feature within the app constantly shares location with other users or uses the Geofencing API. One possible example is Within a family location sharing app, a feature allows users to continuously share location with family members.

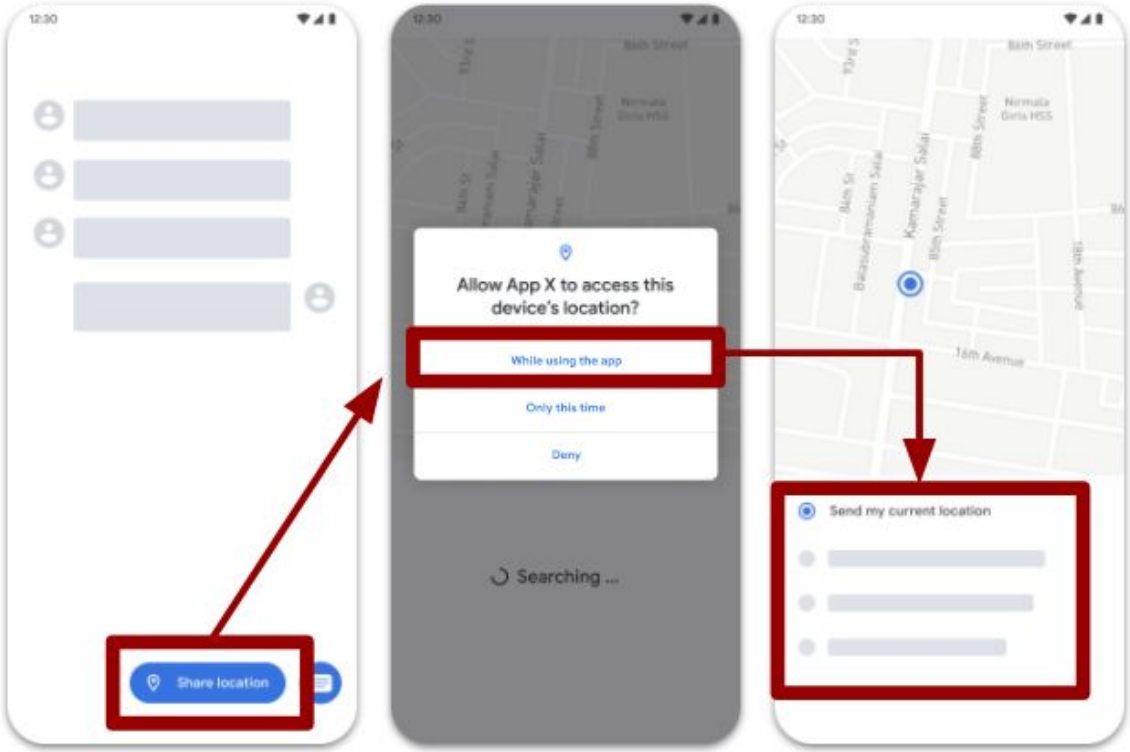
The system considers your app to be using background location if it accesses the device's current location in any situation other than the ones described in the foreground location section

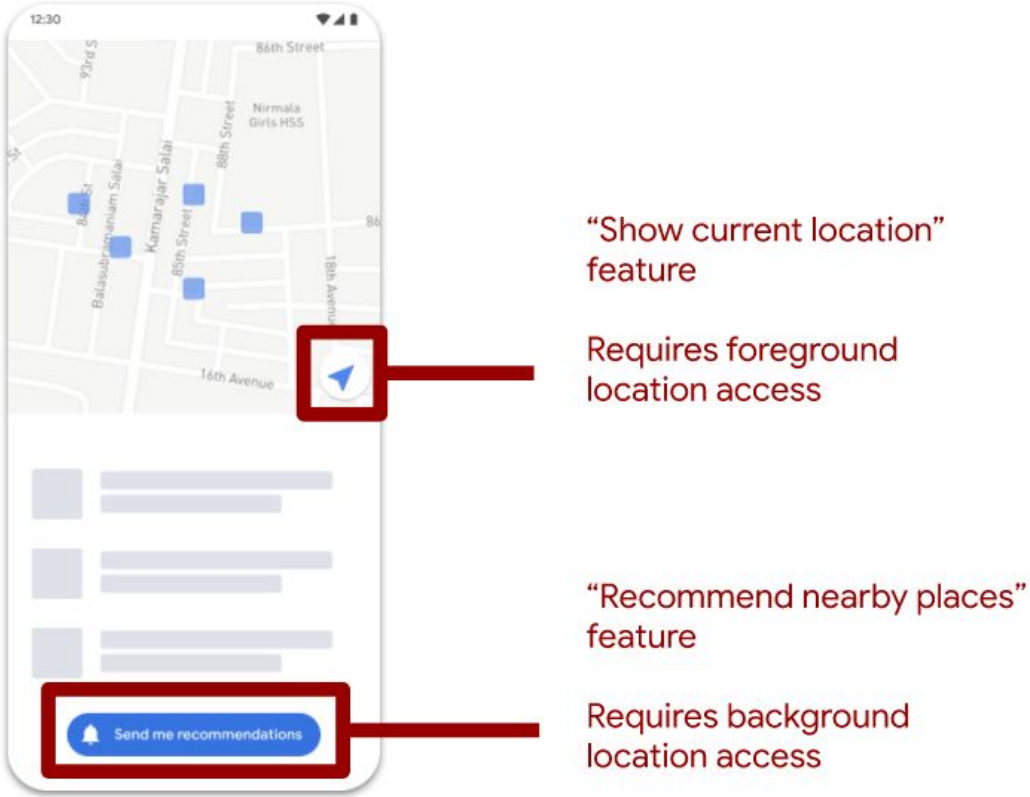
Background Location Continued...

On Android 10 (API level 29) and higher, you must declare the ACCESS_BACKGROUND_LOCATION permission in your app's manifest in order to request background location access at runtime. On earlier versions of Android, when your app receives foreground location access, it automatically receives background location access as well.

```
<manifest ... >
  <!-- Required only when requesting background location access on
       Android 10 (API level 29) and higher. -->
  <uses-permission
android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
</manifest>
```

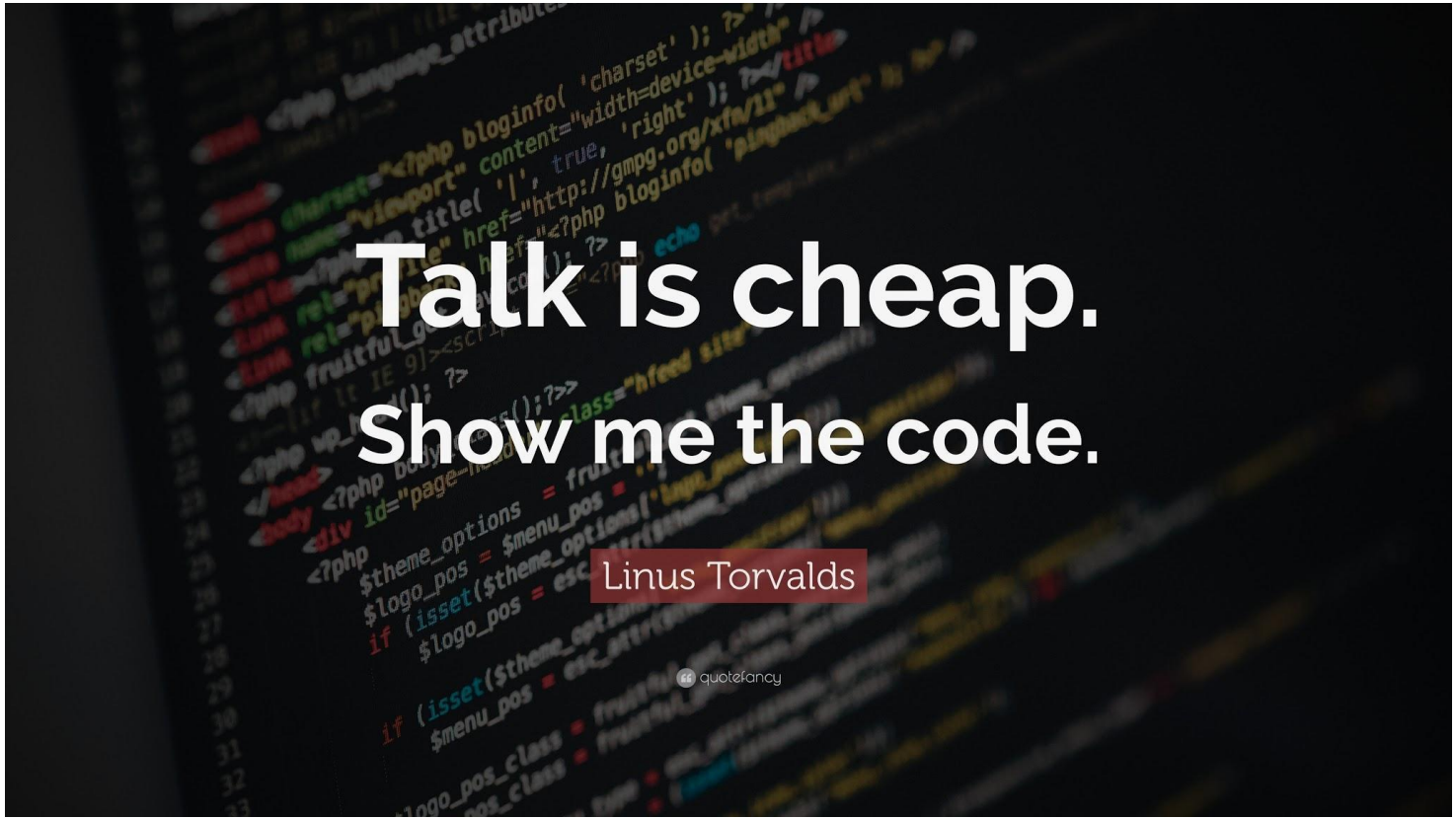
Request location at runtime





If your app targets Android 11 (API level 30) or higher, the system enforces this best practice. If you request a foreground location permission and the background location permission at the same time, the system ignores the request and doesn't grant your app either permission.

There are only 10 types of people in this world. Those who know binary and those who don't.



Thank You!