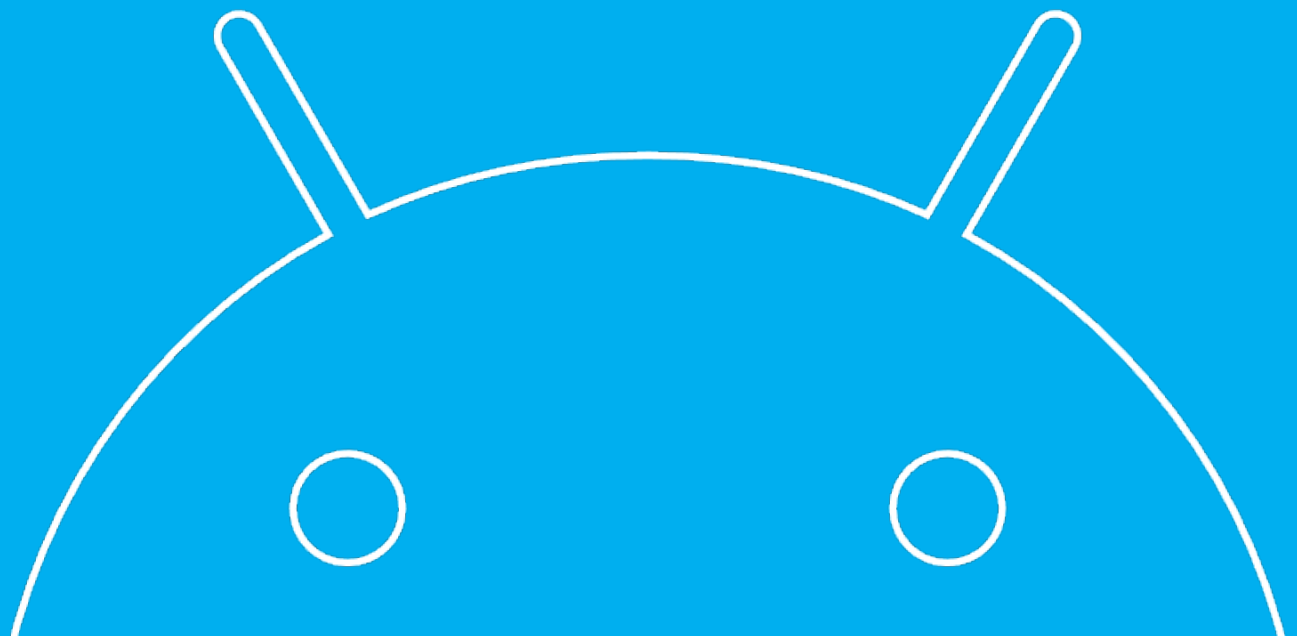


Android Tech Report

The latest news and updates

Vol. 1 (2022 Edition)



Content



Android OS

03 - 10



Google Play

11 - 13



App Center

14 - 15



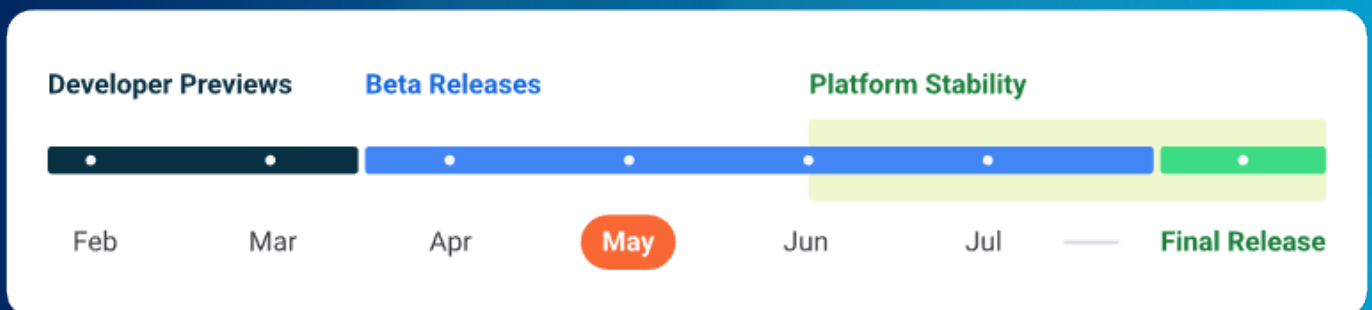
Firebase

16 - 20



Android OS

Android 13 Beta is available to users and developers, so you should test your app's compatibility and release any needed updates. Developers should continue compatibility testing and watch for feedback from Android Beta users. Start early testing by targeting Android 13. Notify SDK and library developers of any issues.



You can test Beta 2 on some of the Android partners' devices:

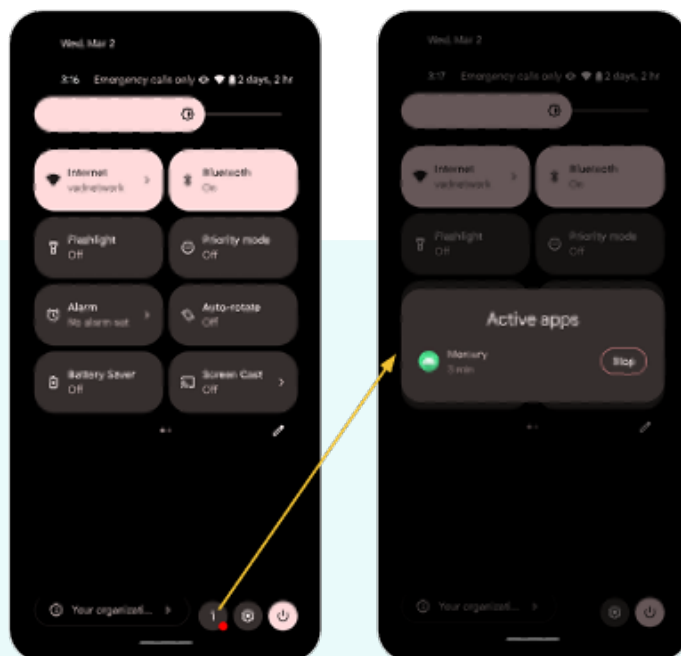
Nokia, Asus, Google Pixel, Lenovo, One Plus, Oppo, Realme, Sharp, Tecno, Vivo, Xiaomi, ZTE.

Android 13 Beta 2

Behavior changes for all apps installed on devices with Android 13. The following behavior changes apply to all apps running on Android 13, regardless of the target SDK version. You should test your app and then modify it as needed to support these properly, where applicable.

[Read more](#)

Performance and Battery



Foreground Services (FGS) Task Manager

Android 13 allows users to stop [foreground services](#) from the [notification drawer](#), regardless of the target SDK version. This new affordance, called the FGS Task Manager, shows a list of apps currently running a foreground service. This list is labeled Active apps. Next to each app is a Stop button.

[Read more](#)

JobScheduler Prefetch Config

JobScheduler provides a way for apps to mark specific jobs as "prefetch" jobs (`JobInfo.Builder.setPrefetch()`), meaning that they should ideally run close to, and before, the next app launch to improve user experience. Historically, JobScheduler has only used the signal to let prefetch jobs exploit free or excess data.

[Read more](#)

Battery Resource Utilization

Better device battery life management

Updated rules on when the system places your app in the "restricted" [App Standby Bucket](#). New limitations on the work that your app can do when the user places your app in the "restricted" state for background battery usage. New system notifications that warn users about long-running [foreground services](#).

[Read more](#)

High Priority Firebase Cloud Message (FCM) Quotas

App Standby Buckets no longer determine how many high-priority FCMs an app can use. High priority FCM quotas scale in proportion to the number of notifications shown to the user in response to High Priority FCMs.

[Read more](#)

Privacy

Runtime Permissions for Notifications

Android 13 introduces a new runtime [notification permission: POST NOTIFICATIONS](#). This change helps users focus on the notifications that are most important to them.

User Experience

In-app Language Pickers

Android 13 adds a centralized location in phone settings for setting per-app language preferences. If your app supports multiple languages, we strongly recommend that you [declare the android: locale Config attribute in your app's manifest](#) so that users can change your app's language settings in the same place that they change language settings for other apps. Additionally, apps that currently use custom in-app language pickers should use the [new APIs for per-app language preferences](#) instead.

Core Functionality

Legacy Copy of Speech Service Implementation Removed

Android 13 removes the SpeechService implementation – including Voice IME, [RecognitionService](#) and an [intent-based API](#) – from the Google app.



Behaviour changes

for apps installed on a device with Android 13 and targeting this version:

Privacy

Notification permission affects foreground service appearance

If the user denies the notification permission, they still see notices related to these foreground services in the [Foreground Services \(FGS\) Task Manager](#). Still, they don't see them in the [notification drawer](#).

New runtime permission for nearby Wi-Fi devices

Because it's difficult for users to associate location permissions with Wi-Fi functionality, Android 13 introduces new runtime permission in the [NEARBY_DEVICES](#) permission group for apps that manage a device's connections to nearby access points over Wi-Fi. This permission, [NEARBY_WIFI_DEVICES](#), fulfills these Wi-Fi use cases.

As long as your app doesn't derive physical location information from the Wi-Fi APIs, request [NEARBY_WIFI_DEVICES](#) instead of [ACCESS_FINE_LOCATION](#) when you target Android 13 or higher and use Wi-Fi APIs. This process is similar to the one you do in Android 12 (API level 31) and higher when you [assert that Bluetooth device information is never used for location](#).

Granular media permissions

If your app targets Android 13, you must request one or more new permissions instead of the [READ_EXTERNAL_STORAGE](#) and [WRITE_EXTERNAL_STORAGE](#) permissions. If the user previously granted your app the [READ_EXTERNAL_STORAGE](#) permission, the system automatically grants each further permission to your app.

Security

Intent Filters Block Non-matching Intents

When your app sends an intent to an exported component of another app that targets Android 13 or higher, that intent is delivered if and only if it matches an `<intent-filter>` element in the receiving app. Non-matching intents are blocked if receiving app upgrades to Android 13 or higher. All intents originating from external apps are delivered to an exported component if and only if it matches its declared `<intent-filter>` elements, regardless of the sending app's target SDK version.

Accessibility

Audio Description

Android 13 introduces a new system-wide accessibility preference that allows users to enable audio descriptions across all apps. An audio description is an additional narration track that consists of a narrator talking through the presentation, describing what is happening on the screen during natural pauses in the audio.

Core Functionality

OpenJDK 11 Updates

Support for the `var` keyword for local variables and as parameters lambdas, new methods in `String` class.

[Read more](#)

Developers' Productivity

Android 13 introduces great new features and APIs for developers. The sections below will help you learn about new features and how to get started with the related APIs.

Predictive Back Gesture

Android 13 introduces a predictive back gesture for Android devices such as phones, large screens, and foldables. Supporting this feature requires you to update your app. To see detailed documentation, see [Update your app to support a predictive back gesture](#). You can also try our codelab.

Themed App Icons

Starting with Android 13, users can opt into themed app icons. With this feature, app icons in supported Android launchers are tinted to inherit the coloring of the user's chosen wallpaper and other themes.

[Read more](#)

Pre-app Language Preferences

In many cases, multilingual users set their system language to one, such as English. Still, they want to select other languages for specific apps, such as Dutch, Chinese, or Hindi. To help apps provide a better experience for these users, Android 13 introduces the following features for apps that support multiple languages: System settings that let users select a preferred language for each app in a centralized location. APIs that let apps set a different language at runtime to use in their UI.

Color Vector Fonts

Android 13 adds rendering support for COLR version 1 (COLRv1) fonts and updates the system emoji to the COLRv1 format. COLRv1 is a new and highly compact font format that renders quickly and crisply at any size. For most apps, the system handles everything, and it just works. However, if your app implements its own text rendering and uses the system's fonts, we recommend testing emoji rendering.

Privacy and Security

Safer Exporting of Context-registered Receivers

To help make runtime receivers safer, Android 13 allows you to specify whether a particular broadcast receiver in your app should be exported and visible to other apps on the device.

If the broadcast receiver is exported, other apps could send unprotected broadcasts to your app.

New Permission to Use Exact Alarms

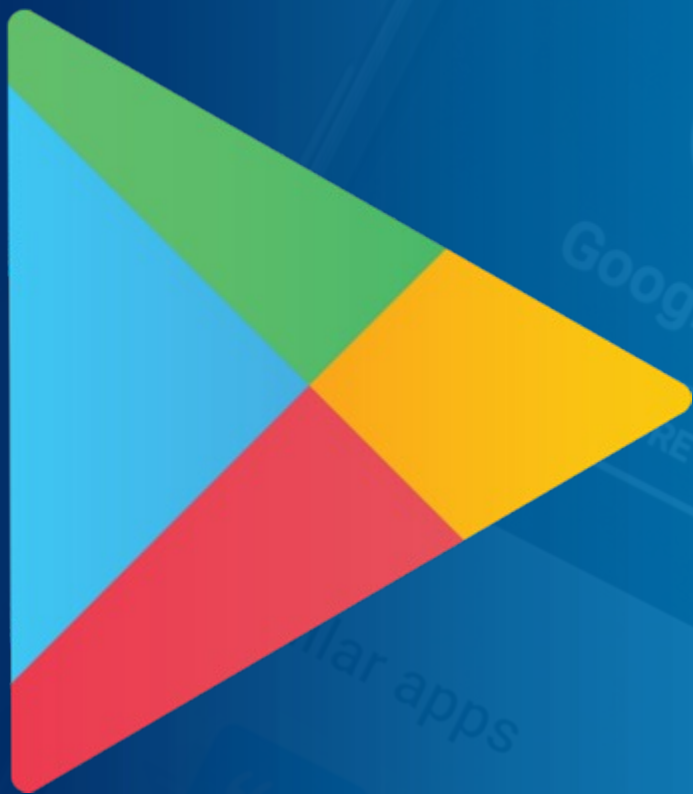
If your app targets Android 13, you can use the [Use Exact Alarm](#) permission, which is automatically granted to your app. In order for your app to use this permission, however, it must satisfy at least one of the following criteria: Your app is an alarm app or a clock app; Your app is a calendar app that shows notifications for upcoming events.

Photo Picker

Android 13 includes a new [photo picker](#) experience. When the app launches the photo picker, users select specific images and videos to share with your app, such as profile pictures, instead of giving your app access to view the entire media library.

Developer Downgradable Permissions

Starting in Android 13, your app can revoke access to runtime permissions previously granted by the system or by the user.



Google Play

Upcoming Google Play Deadlines

11.05.
2022

[Families' policy](#)
[Hate speech policy](#)
[Deceptive behavior - Misleading claim policy](#)
[Families certified Ad SDK Program policy](#)
[Accessibility policy](#)

11.07.
2022

[Request install packages permission policy](#)

11.08.
2022

[News policy](#)

03.10.
2022

[Health connect by Android permissions policy](#)
[User data policy](#)

01.11.
2022

[Target API level requirements](#)

Past Deadlines

15.12.
2021

[Subscriptions policy](#)

18.12.
2021

[Payments policy](#)

Google Play Important Updates

Android Emoji Policy

Google is clarifying existing language in the Android Emoji policy about which apps are already compliant and which apps will need to come into compliance.

[Read more](#)

Enabling Dishonest Behavior Policy

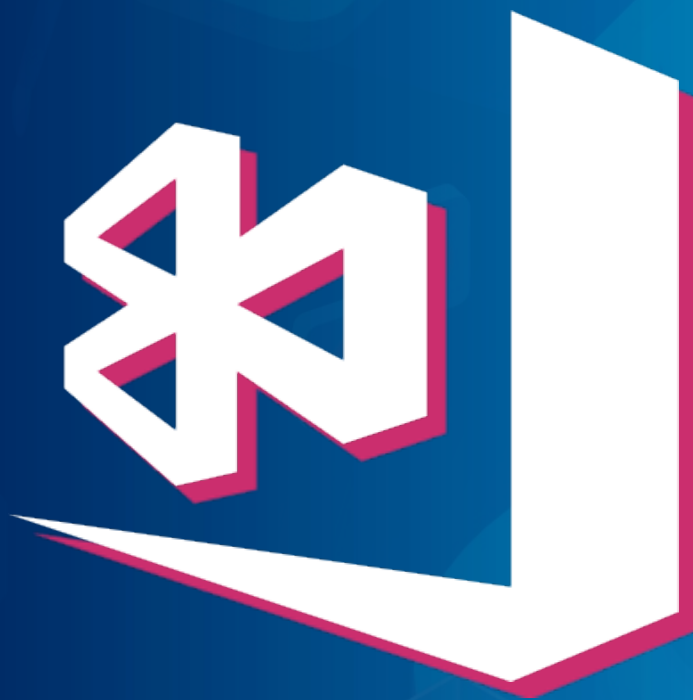
Google clarified in Enabling Dishonest Behavior policy that it prohibits apps that generate fake bank accounts.

[Read more](#)

Family Policy

Clarifying existing language in the Families policy to make it clear that apps that solely target children may not request location permission or collect, use, and transmit [precise location](#).

[Read more](#)



App Center

Public Preview for App Center Build Export

New capability to migrate your existing App Center Build configurations over to Azure Pipelines. You can export your pipelines in a form of a JSON or a YAML file.

[Read more](#)

App Center Analytics & Diagnostics Data Retention

Starting June 30, 2022, Analytics and Diagnostics data retention will be limited to 28 days.

[Read more](#)

Deprecating macOS 10.14 image in App Center Build

If you are building iOS or Xamarin apps using Xcode 10.3, Xamarin.iOS 12.14 or Xamarin.Android 9.4 or below versions, you will be impacted by this change.

[Read more](#)

Run Tests on Android and iOS Beta Versions

App Center offers you a way to try pre-release versions of Android and iOS prior to the final release.

[Read more](#)

Apache Cordova Retirement

Starting April 1, 2022, the App Center service will no longer accept calls from the Cordova SDK.

[Read more](#)



Firebase

New Extension Events Enable Additional Customization

Firebase Extensions are pre-packaged bundles of code that make it easier to add core features to apps from Google and other trusted companies with minimal code and without learning many different third-party APIs.

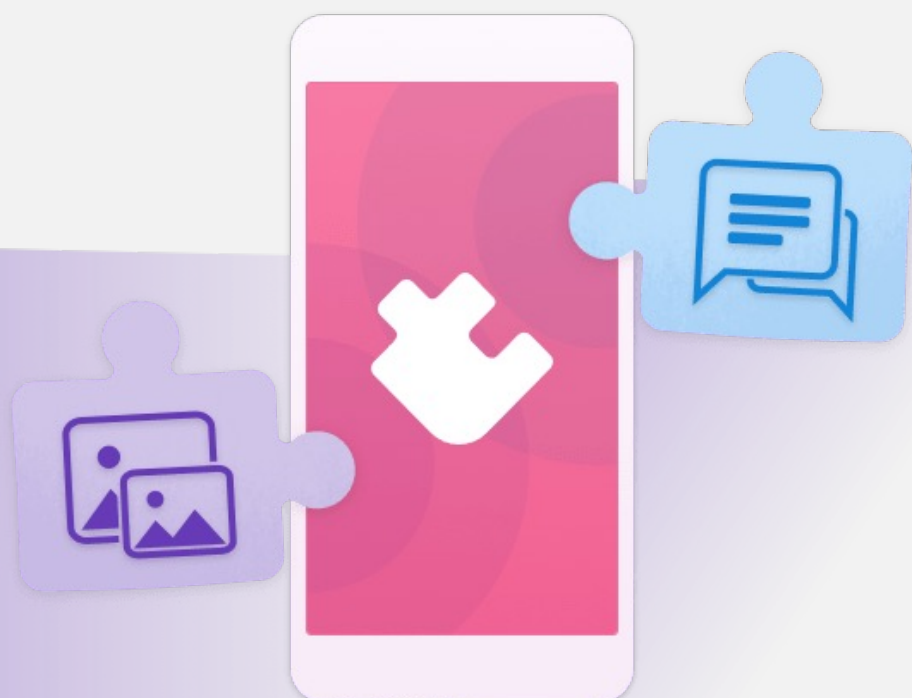
Firebase announces the release of third-party extensions for marketing, search, and payment processing. Two new extensions from [Snap](#) allow users to log in with Snap and promote products in their stories. With new [Stream](#) extensions, developers can implement chat in apps and respond to users in real-time.

[RevenueCat](#) helps manage in-app

subscriptions, while [Typesense](#) is an open-source option for implementing indexing and search in Cloud Firestore.

Moreover, all Extensions are now available in the Emulator Suite, so they can be tested in a safe local environment to understand their capabilities better and make adjustments quickly. To install Extensions, go to the [Extensions page](#) on the Firebase website or in the Firebase console.

[Read more](#)



Firebase Performance Monitoring

Firebase Performance Monitoring gathers and presents data about the app's performance. This way, developers can know exactly what's happening in their apps and when users are experiencing slow performance.

A few months ago, Firebase released performance alerts to inform developers of latency issues affecting app start time. As part of this feature, graduating from beta to general availability, you can set up new alerts for custom code traces, network requests, screen renderings, and web traces for the most

critical parts of your app – and customize them to your own thresholds.

With performance alerts, apps can be released with confidence, knowing that when users are experiencing issues, you'll be the first to know. Configure your alerts by following the instructions in the Firebase [documentation](#).

[Read more](#)

App Check Security Enhancements

Firebase App Check protects backend infrastructure from threats like billing fraud, phishing, app impersonation, data theft and poisoning. It does this by attesting that incoming traffic is coming from your app on a legitimate device and blocking traffic that doesn't have valid credentials.

Firebase also announced that App Check integrates with the Play Integrity API, the latest and most advanced app

attestation provider for Android devices with more anti-abuse signals. In addition, App Check has now graduated from beta to general availability and is certified under major compliance and security standards. To learn more about App Check, check out the [documentation](#).

Read more

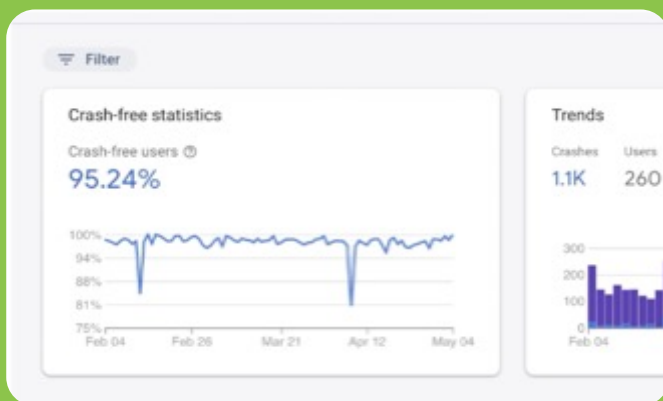


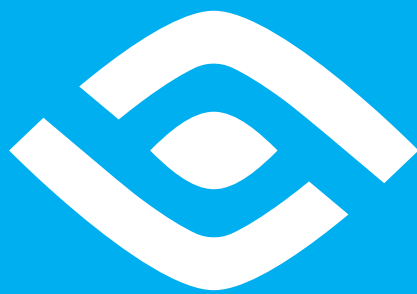
Crashlytics

[Firebase Crashlytics](#) gives a complete view into the app's stability so bugs can be tracked, prioritized, and fixed before they impact a large number of users. However, even with detailed and real-time crash reports, troubleshooting usually requires jumping between the Crashlytics console and IDE to locate and debug the error. This can be a frustrating experience because it's difficult to look at stack traces in a browser and then navigate

through code to find the right place to set breakpoint. That's why Firebase is introducing the new App Quality Insights window in Android Studio, which helps discover, investigate, and reproduce issues reported by Crashlytics within the context of the local Android Studio project. It will also highlight problematic lines of code that have appeared in crash reports.

On top of that, Crashlytics now integrates with Google Play so developers can filter Crashlytics crash reports based on Play Tracks, making it easier to distinguish which crash events are from internal testing vs. open testing vs. in production. In order to use this feature, go to the Crashlytics dashboard and link the app to Google Play.





Thank you!

www.scalefocus.com