

Create an instant-enabled app bundle



Warning: Google Play Instant will no longer be available. Starting December 2025, Instant Apps cannot be published through Google Play, and all [Google Play services Instant APIs](https://developers.google.com/android/reference/com/google/android/gms/instantapps/package-summary) (<https://developers.google.com/android/reference/com/google/android/gms/instantapps/package-summary>)

will no longer work. Users will no longer be served Instant Apps by Play using any mechanism.

We're making this change based on developer feedback and our continuous investments to improve the ecosystem since the introduction of Google Play Instant.

To continue optimizing for user growth, we encourage developers to refer users to their regular app or game, using [deeplinks](https://support.google.com/googleplay/android-developer/answer/12463044) (<https://support.google.com/googleplay/android-developer/answer/12463044>) to redirect them to specific journeys or features when relevant.

You create a Google Play Instant experience by including it as part of an [Android App Bundle](/guide/app-bundle) (</guide/app-bundle>). Such a bundle is known as an *instant-enabled app bundle*. This document shows you how to set up your development environment for instant-enabled app bundles, as well as how to configure, build, test, and publish an instant-enabled app bundle.

If you have an existing instant app project that uses the deprecated feature plugin (`com.android.feature`), learn how to [migrate your instant app to support Android App Bundles](/topic/google-play-instant/feature-module-migration) (</topic/google-play-instant/feature-module-migration>).

Note: In order to publish an app bundle, you must opt into [Play App Signing](https://support.google.com/googleplay/android-developer/answer/7384423) (<https://support.google.com/googleplay/android-developer/answer/7384423>).

Set up development environment

To provide an instant experience within an app bundle, you need access to the Google Play Instant Development SDK. You can install the SDK using one of the following methods:

- [Install Android Studio 3.6](#) (/studio) or higher. After opening Android Studio, download the Google Play Instant Development SDK from the **SDK Tools** tab in the [SDK Manager](#) (/studio/intro/update#sdk-manager).
- Install from the command line:

```
$ cd path/to/android/sdk /tools/bin && \  
./sdkmanager 'extras;google;instantapps'
```

Additionally, if you want to test your instant experience locally, get access to either a physical or [virtual](#) (/studio/run/managing-avds) device.

Learn about the required execution conditions

Google Play Instant runs instant-enabled app bundles in a special kind of SELinux sandbox for added security. This sandbox permits a subset of permissions, as well as limited types of interactions with other apps. The following sections explain the characteristics of this sandbox in more detail.

Supported permissions and operations

Instant-enabled app bundles can only use permissions from the following list:

- **[ACCESS_COARSE_LOCATION](#)** (/reference/android/Manifest.permission#ACCESS_COARSE_LOCATION)
- **[ACCESS_FINE_LOCATION](#)** (/reference/android/Manifest.permission#ACCESS_FINE_LOCATION)
- **[ACCESS_NETWORK_STATE](#)** (/reference/android/Manifest.permission#ACCESS_NETWORK_STATE)
- **BILLING** – *Deprecated as of [Play Billing Library 1.0](#)* (/google/play/billing/billing_library_releases_notes#release-1_0).
- **[CAMERA](#)** (/reference/android/Manifest.permission#CAMERA)
- **[INSTANT_APP_FOREGROUND_SERVICE](#)** (/reference/android/Manifest.permission#INSTANT_APP_FOREGROUND_SERVICE) – *Only in*

Android 8.0 (API level 26) and higher.

- **INTERNET** (/reference/android/Manifest.permission#INTERNET)
- **READ_PHONE_NUMBERS** (/reference/android/Manifest.permission#READ_PHONE_NUMBERS) – *Only in Android 8.0 (API level 26) and higher.*
- **RECORD_AUDIO** (/reference/android/Manifest.permission#RECORD_AUDIO)
- **VIBRATE** (/reference/android/Manifest.permission#VIBRATE)
- **WAKE_LOCK** (/reference/android/Manifest.permission#WAKE_LOCK)

Handling common unsupported permissions

The following is a list of common, non-supported permissions that you must remove from your application and the recommended migration path for each:

- **ACCESS_WIFI_STATE**: Use **ACCESS_NETWORK_STATE**, which should provides information similar to **ACCESS_WIFI_STATE**.
- **BILLING**: This is a deprecated permission. Use the [Google Play Billing Library](#) (/google/play/billing/billing_library_overview), which no longer requires the **com.android.vending.BILLING** permission.
- **READ/WRITE_EXTERNAL_STORAGE**: Instant apps do not have access to external storage; use internal storage instead.
- **com.google.android.c2dm.permission.RECEIVE** and **permission.C2D_MESSAGE**: C2DM is deprecated. Migrate to Firebase Cloud Messaging (FCM). FCM does not need any additional permissions to work.

In addition, instant-enabled app bundles cannot do the following:

- Use [background services](#) (/training/run-background-service/create-service).
- [Send notifications](#) (/training/notify-user/build-notification) when running in the background.

Access to installed apps

When developing an instant experience, keep in mind that it cannot interact with installed apps on a device unless one of the following is true:

- One or more activities within an installed app has set its `android:visibleToInstantApps` element to `true` – *This element is available to apps running Android 8.0 (API level 26) or higher.*
- An installed app contains an intent filter that includes `CATEGORY_BROWSABLE` ([/reference/android/content/Intent#CATEGORY_BROWSABLE](#)).
- The instant experience is sending an intent using either the `ACTION_SEND` ([/reference/android/content/Intent#ACTION_SEND](#)), `ACTION_SENDTO` ([/reference/android/content/Intent#ACTION_SENDTO](#)), or `ACTION_SEND_MULTIPLE` ([/reference/android/content/Intent#ACTION_SEND_MULTIPLE](#)) action.

Configure your project for instant experiences

To be compatible with Google Play Instant, you must configure several aspects of your instant-enabled app bundle carefully. The following sections describe these considerations.

Declare project dependencies

To use the Google Play Instant APIs in your app, include the following declaration in your app module's `build.gradle` configuration file:

```
GroovyKotlin (#kotlin)
(#groovy)
```

```
implementation "com.google.android.gms:play-services-instantapps:17.0.0"
```

Define the correct version codes

The version code of your app's instant experience needs to be less than the version code of the installable app. The expectation is that users move from the Google Play Instant experience to downloading and installing the app onto their device. The Android framework considers this transition to be an app update.

Note: If the user has the installed version of your app on their device, that installed version always runs instead of your instant experience. This is true even if the installed version is an older version of your app compared to your instant experience.

To make sure that you follow the versioning scheme that users expect, follow one of these strategies:

- Restart the version codes for the Google Play Instant experience at 1.
- Increase the version code of the installable APK by a large number, such as 1000, to ensure that there is enough space for your instant experience's version number to increase.

It's OK to develop your instant app and your installable app in two separate Android Studio projects. If you do so, however, you must do the following to publish your app on Google Play:

1. Use the same package name in both Android Studio projects.
2. In the Google Play Console, upload both variants to the same application.

Note: The version code isn't a user-facing value and is primarily used by the system. The user-facing *version name* has no restrictions.

For more details on setting your app's version, see [Version your app](/studio/publish/versioning) (/studio/publish/versioning).

Update the target sandbox version

Your instant app's `AndroidManifest.xml` file needs to be updated to target the sandbox environment that Google Play Instant supports. You can complete this update by adding the `android:targetSandboxVersion` attribute to your app's `<manifest>` element, as shown in the following code snippet:

```
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
```

```
...  
android:targetSandboxVersion="2" ...>
```

For more information, see documentation on the [targetSandboxVersion](/guide/topics/manifest/manifest-element#targetSandboxVersion) (/guide/topics/manifest/manifest-element#targetSandboxVersion) attribute.

Declare instant-enabled app modules

You can declare that your app bundle supports instant experiences using one of the following methods:

- If you have an existing app bundle that contains only a [base module](#) (/studio/projects/dynamic-delivery/configure-base), you can instant-enable the app bundle as follows:
 1. Open the **Project** panel by selecting **View > Tool Windows > Project** from the menu bar.
 2. Right-click on your base module, typically named 'app', and select **Refactor > Enable Instant Apps Support**.
 3. In the dialog that appears, select your base module from the dropdown menu.
 4. Click **OK**.

Android Studio adds the following declaration to the module's manifest:

```
<manifest ... xmlns:dist="http://schemas.android.com/apk/distribution">  
    <dist:module dist:instant="true" />  
    ...  
</manifest>
```



Note: By default, the base module of an app bundle is called **app**.

- If you have an existing app bundle that contains multiple modules, you can [Create an instant-enabled feature module](/studio/projects/dynamic-delivery/instant-delivery) (/studio/projects/dynamic-delivery/instant-delivery). This process also instant-enables your app's base module, giving you the option to [support multiple instant entry points](/topic/google-play-instant/guides/multiple-entry-points) (/topic/google-play-instant/guides/multiple-entry-points) within your app.

Note: A given module can contain multiple activities. For an app bundle to be instant-enabled, however, the **combined** download size of the code and resources within **all** instant-enabled modules must be at most 15 MB.

Add support for sign-in

If your instant experience allows users to sign in, your instant-enabled app bundle must [support Smart Lock for Passwords on Android](#)

(<https://developers.google.com/identity/smartlock-passwords/android/>). If you are building an ["Instant play" game](/topic/google-play-instant/instant-play-games) (/topic/google-play-instant/instant-play-games), you should use [Google Play Games Services](#) (<https://developers.google.com/games/services/>) sign-in instead.

Support the execution environment

To be compatible with the SELinux sandbox in which instant experiences run, keep the following in mind when creating your instant-enabled app bundle:

- Don't share the value of `myUid()` ([/reference/android/os/Process#myUid\(\)](/reference/android/os/Process#myUid())), which is your app process's kernel-assigned [UID](http://www.linfo.org/uid.html) (<http://www.linfo.org/uid.html>).
- If your app targets Android 8.1 (API level 27) or lower, create a [Network Security Config](#) (/training/articles/security-config) file, and set `cleartextTrafficPermitted` to `false`. Instant experiences don't support HTTP traffic. For apps that target Android 9 or higher, cleartext traffic is disabled by default.
- Your instant experience remains downloaded on a user's device until the instant experience cache is cleared, which occurs in one of the following situations:
 - The instant experience cache is garbage-collected because the device is running low on available memory.
 - The user restarts their device.

If either process occurs, the user must re-download your instant experience in order to interact with it.

- If the system is running very low on storage space, it's possible that your instant experience's user data is removed from internal storage. Therefore, it's recommended to periodically sync user data with your app's server so that the user's progress is preserved.

Add logic for instant experience workflows

After you configure your app bundle so that it supports instant experiences, add the logic that's shown in the following sections.

Check whether app is running instant experience

If some of your app's logic depends on whether the user is engaged in your instant experience, call the `isInstantApp()`

([https://developers.google.com/android/reference/com/google/android/gms/instantapps/PackageManagerCompat.html#isInstantApp\(\)](https://developers.google.com/android/reference/com/google/android/gms/instantapps/PackageManagerCompat.html#isInstantApp()))

method. This method returns `true` if the currently-running process is an instant experience.

Display an install prompt

If you are building a trial version of your app or game, Google Play Instant allows you to display a prompt within your instant experience, inviting users to install the full experience on their device. To display this prompt, use the `InstantApps.showInstallPrompt()`

([https://developers.google.com/android/reference/com/google/android/gms/instantapps/InstantApps.html#showInstallPrompt\(android.app.Activity,%20android.content.Intent,%20int,%20java.lang.String\)](https://developers.google.com/android/reference/com/google/android/gms/instantapps/InstantApps.html#showInstallPrompt(android.app.Activity,%20android.content.Intent,%20int,%20java.lang.String)))

method, as shown in the following code snippet:

KotlinJava (#java)
(#kotlin)

```
class MyInstantExperienceActivity : AppCompatActivity {  
    // ...  
    private fun showInstallPrompt() {  
        val postInstall = Intent(Intent.ACTION_MAIN)
```

```

        .addCategory(Intent.CATEGORY_DEFAULT)
        .setPackage(your-installed-experience-package-name )

        // The request code is passed to startActivityForResult().
        InstantApps.showInstallPrompt(this@MyInstantExperienceActivity,
            postInstall, request-code , /* referrer= */ null)
    }
}

```

Transfer data to an installed experience

If the user enjoys your instant experience, they might decide to install your app. To provide a good user experience, it's important that the user's data is transferred from your instant experience to the full version of your app.

If the user is using a device that runs Android 8.0 (API level 26) or higher, and if your app specifies a `targetSandboxVersion` of 2 (`#target-sandbox-version`), then the user's data is transferred automatically to the full version of your app. Otherwise, you must transfer the data manually. To do so, use one of the following APIs:

- For users who use devices that run Android 8.0 (API level 26) and higher, use the Cookie API - [sample app](#)
(<https://github.com/android/app-bundle-samples/tree/main/InstantApps/cookie-api>)
- If users can interact with your experience on devices that run Android 7.1 (API level 25) and lower, add support for the Storage API - [sample app](#)
(<https://github.com/android/app-bundle-samples/tree/main/InstantApps/storage-api>)

Build the app bundle

You can use either Android Studio or the command-line interface to build your instant-enabled app bundle.

Android Studio

Using Android Studio, you can build your app bundle by selecting **Build > Build Bundle(s) / APK(s) > Build Bundle(s)**. For more information about building your project, see [Build your](#)

[project](#) (/studio/run/build-for-release).

Command-line interface

You can also [build the app bundle from the command line](#) (/studio/build/building-cmdline) using Gradle.

Support 64-bit architectures

Apps published on Google Play need to support 64-bit architectures. Adding a 64-bit version of your app provides performance improvements and sets you up for devices with 64-bit-only hardware. [Learn more about 64-bit support](#) (/google/play/requirements/64-bit).

Test the instant experience

Before publishing your instant-enabled app bundle, you can test the instant experience from one of the following locations to verify functionality:

- Install onto a local device using [Android Studio](#) (#test-from-studio).
- Install onto a local device using the [command-line interface](#) (#test-from-cli).
- Publish to the [internal test track](#) (#test-publish-internal-test-track) on the Google Play Console.

Note: Instant experiences are available only on devices running Android 8.0 (API level 26) or higher.

Android Studio

To test your app's instant experience on a local machine using Android Studio, complete the following steps:

1. If you have an installed version of your app on your test device, uninstall it.

2. In Android Studio, from the installation options that appear on the **General** tab of the Run/Debug Configurations dialog (/studio/run/rundebbugconfig#opening), enable the **Deploy as instant app** checkbox.
3. Select **Run > Run** in the menu bar, or click **Run**  in the toolbar, then choose the device where you'd like to test your app's instant experiences. Your app's instant experience loads on the test device that you've chosen.

Command-line interface

To test your app's instant experience on a local machine using the command line, complete the following steps:

1. If you have an installed version of your app on your test device, uninstall it.
2. Sideload and run your instant app on your test device by entering the following command:

```
$ ia run output-from-build-command 
```

Note: You can provide alternative information to **ia run** to launch an instant experience on a test device. For more information, execute the command: **ia help run**.

Internal test track

To test your app's instant experience from the Play Store or a banner on your website, publish the app to the internal test track (https://support.google.com/googleplay/android-developer/answer/3131213#internal_test) on the Play Console.

To publish your app to the internal test track, complete the following steps:

1. Upload your app bundle by following the steps in the Upload your app bundle to the Play Console (/studio/publish/upload-bundle) guide.

2. Prepare the uploaded bundle for a release to the internal test track. For more information, see the support article on how to [Prepare & roll out releases](https://support.google.com/googleplay/android-developer/answer/7159011) (<https://support.google.com/googleplay/android-developer/answer/7159011>).
3. Sign into an internal tester account on a device, then launch your instant experience from one of the following surfaces:
 - The **Try Now** button from your app's Play Store listing.
 - A link from a banner on your app's website.

Note: When you use Google Play's internal test track, the Play Console doesn't apply the size limits discussed in the Google Play Instant overview. Therefore, you can test and internally showcase an instant experience, even if it's larger than 15 MB.

Publish the app bundle to the production track

Note: The [Designed for Families program](/google-play/guides/families) (/google-play/guides/families) is not currently compatible with Google Play Instant. If your app or game is not designed for children you will need to [disenroll from the Designed for Families program](#) (<https://support.google.com/googleplay/android-developer/answer/9285070>) in order to publish an instant app or game.

Note: From June 2023 on, only instant apps published using app bundles are available to users. Please ensure all APK based instant apps have been updated to instant enabled bundles.

To publish your instant-enabled app bundle, complete the following steps:

1. If you haven't already, [sign your app bundle with a release key](#) (/studio/publish/app-signing#sign-apk) and [upload the app bundle to the Play Console](#) (/studio/publish/upload-bundle).

2. In the Play Console, open **Release management > Android Instant Apps**, then navigate to the **instant app production** track.
3. Select **Update from Library**, then select the instant-enabled app bundle that you've uploaded.

Choose where to publish your instant experience

It's possible to launch an instant experience of your app in a subset of the countries and regions where people can install your app. This capability is useful in cases where you want to promote your app's instant experience to users who reside in a specific set of countries and regions.

Additional resources

To learn more about creating instant experiences and Android App Bundles, see the following resources:

Video: Bundling an App in an Instant (<https://youtu.be/L9J2e5PYXNg>)

Learn how to add an instant experience to an Android App Bundle in this session from Android Dev Summit '18.

Video: Publish smaller apps with the Android App Bundle (<https://youtu.be/9D63S4ZRBIs>)

Learn how app bundles help you develop your app more quickly and create smaller APKs for your users.

Codelab: Your First Android App Bundle

(<https://codelabs.developers.google.com/codelabs/your-first-dynamic-app/index.html>)

A step-by-step guide for creating an Android App Bundle and adding features to it.

The Android App Bundle Format (/guide/app-bundle/build#aab_format)

Learn more about how the `bundletool` command-line program organizes an app bundle from your app's code and resources.

Content and code samples on this page are subject to the licenses described in the [Content License \(/license\)](#).
Java and OpenJDK are trademarks or registered trademarks of Oracle and/or its affiliates.

Last updated 2026-06-24 UTC.