

android developer api download



Get the Android SDK.

Before installing the Android SDK, you must agree to the following terms and conditions.

Terms and Conditions.

1. Introduction.
2. Accepting this License Agreement.
3. SDK License from Google.
4. Use of the SDK by You.
5. Your Developer Credentials.
6. Privacy and Information.
7. Third Party Applications.
8. Using Android APIs.
9. Terminating this License Agreement.
10. DISCLAIMER OF WARRANTIES.
11. LIMITATION OF LIABILITY.
12. Indemnification.
13. Changes to the License Agreement.
14. General Legal Terms.

Now that you've downloaded the Android SDK, you don't need to return here for SDK updates. The SDK tools allow you to install additional packages and future updates from the SDK Manager.

For instructions about setting up your Android SDK for the first time, read [Setting Up the ADT Bundle](#).

I have read and agree with the above terms and conditions.

Get the Android SDK.

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in ADT (Android Developer Tools) to streamline your Android app development.

With a single download, the ADT Bundle includes everything you need to begin developing apps:

Eclipse + ADT plugin Android SDK Tools Android Platform-tools The latest Android platform The latest Android system image for the emulator.

To get the ADT Bundle or stand-alone SDK Tools, please visit the web site at developer.android.com/sdk/

Choose the SDK package for your OS from the table below.

If you prefer to use an existing version of Eclipse or another IDE, you can instead take a more customized approach to installing the Android SDK. See the following instructions.

USE AN EXISTING IDE.

If you already have an IDE you want to use for Android app development, setting up a new SDK requires that you download the SDK Tools, then select additional Android SDK packages to install (such as the Android platform and system image). If you'll be using an existing version of Eclipse, then you can add the ADT plugin to it.

SYSTEM REQUIREMENTS.

Operating Systems.

Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit) Mac OS X 10.5.8 or later (x86 only) Linux (tested on Ubuntu Linux,

Lucid Lynx) GNU C Library (glibc) 2.7 or later is required. On Ubuntu Linux, version 8.04 or later is required. 64-bit distributions must be capable of running 32-bit applications.

Eclipse IDE.

3.6.2 (Helios) or greater.

Note: Eclipse 3.5 (Galileo) is no longer supported with the latest version of ADT.

Other development environments.

(JRE alone is not sufficient) 1.8 or later Not compatible with Gnu Compiler for Java (gcj)

Note: Some Linux distributions may include JDK 1.4 or Gnu Compiler for Java, both of which are not supported for Android development.

Install Android Development Tools.

This 15-minute tutorial shows you how to install Android Studio, which includes the Android SDK, and create an Android Virtual Device (AVD) on which you install an Oracle JavaScript Extension Toolkit (Oracle JET) hybrid mobile application during a later tutorial. The time to complete doesn't include processing time as a result of your activities.

Background.

The Oracle JET command-line interface invokes the Android SDK that you install with Android Studio to build an Android application package (APK) file from the source files of your hybrid mobile application. This APK file is installed on an AVD to enable you to test the hybrid mobile application.

What Do You Need?

A computer that meets the system requirements to install Android Studio. See System requirements at <https://developer.android.com/studio/>

Install and Set Up Android Studio.

Depending on the operating system of your computer, the Android Studio installation wizard prompts you with dialogs where you choose between standard or custom install types. Choose the options recommended by the Android Studio installation wizard as these options include the components that you require to create and install a hybrid mobile application on an AVD. They also include an Android emulator and an emulator accelerator appropriate for your computer, be that Windows, Mac, or Linux.

Go to the Download page for Android Studio on the Android Developer's website at <https://developer.android.com/studio/> and click **DOWNLOAD ANDROID STUDIO** . Review and accept the terms and conditions in the Download Android Studio dialog that appears and click **DOWNLOAD ANDROID STUDIO FOR PLATFORM** where PLATFORM refers to the operating system of your machine, such as Windows, Mac, or Linux. Go to the Install Android Studio guide at <https://developer.android.com/studio/install> and follow the instructions for your operating system to install and start Android Studio. If you run Android Studio behind a firewall or secure network, an Android Studio First Run dialog appears which displays a button (Setup Proxy) that you click to enter the proxy server settings for your environment. This allows Android Studio to complete the download of the components for a standard install. In the Welcome to Android Studio dialog, select Start a new Android Studio project, and then, in the Create New Project wizard, accept the default options to progress to the final screen in the wizard and click Finish .

This enables the Android Studio toolbar with the options that you need to create an AVD. It also downloads and configures the Gradle build tool that the Android SDK invokes to build the APK file when you complete the creation of your Oracle JET hybrid mobile application.

If you run Android Studio behind a firewall or secure network, a Proxy Settings dialog appears that enables you to configure proxy settings for the Gradle build tool that Android Studio downloads and configures. Provide the proxy settings for your environment to ensure that the Gradle build tool functions correctly.

Install an Android SDK Platform Package.

In the Android Studio toolbar, click the SDK Manager icon () to open the Android SDK page in the Default Settings dialog. In the SDK Platforms tab, select Android 8.0 (Oreo) with an API Level value of 26, and click Apply . Description of the illustration [sdkmgrdialog.png](#) Click OK in the confirmation dialog that appears and accept the license agreement to proceed with the installation. Once the installation completes, click Finish and then click OK to close the Default Settings dialog.

Create and Start an Android Virtual Device.

In the Android Studio toolbar, click the AVD Manager icon () and click Create Virtual Device in the Android Virtual Device Manager dialog that opens. In the Phone category of the Choose a device definition page of the Virtual Device Configuration dialog, select Nexus 5X and click Next . In the Recommended tab of the Select a system image page, click Download for the Oreo entry with an API Level of 26. Description of the illustration [downloadoreo.png](#) The SDK Quickfix Installation wizard opens. Accept the license agreement to proceed with the download of the system image for Android 8.0 with API Level 26. Once the installation completes, click Finish and then click Next in the Select a system page. In

the Verify Configuration page, review and accept the default settings such as the AVD Name value of Nexus 5X API 26, then click Finish . In the Android Virtual Device Manager dialog, click the Launch this AVD in the emulator icon () under the Actions column for the newly-created Nexus 5X API 26 AVD.

The Android emulator starts and loads the Nexus 5X API 26 AVD. A toolbar appears to the right of the AVD that provides UI controls to interact with the AVD.

Install the Java SE Development Kit 8.

Go to the Java SE Development Kit 8 Downloads page at <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>, accept the license agreement, and download the Java SE Development Kit installation file for your platform. Follow the JDK installation instructions for your platform at <https://docs.oracle.com/javase/8/docs/technotes/guides/install/toc.html>.

Create Environment Variables.

To function correctly, the Apache Cordova command-line interface that the Oracle JET command-line interface communicates with requires that you configure environment variables. These environment variables reference the installation location of the Android SDK that is installed by Android Studio and the Java SE Development Kit 8.

YouTube Android Player API.

The YouTube Android Player API enables you to incorporate video playback functionality into your Android applications. The API defines methods for loading and playing YouTube videos (and playlists) and for customizing and controlling the video playback experience.

Using the API, you can load or cue videos into a player view embedded in your application's UI. You can then control playback programmatically. For example, you can play, pause, or seek to a specific point in the currently loaded video.

You can also register event listeners to get callbacks for certain events, such as the player loading a video or the player state changing. Finally, the API has helper functionality to support orientation changes as well as transitions to fullscreen playback.

How it works.

The API client library interacts with a service that is distributed as a part of the YouTube app for the Android platform. The client library has a light footprint, meaning it won't adversely impact your app's file size, if you use ProGuard as part of your build process.

As the API develops, you will be able to access newly introduced API features by upgrading to a newer version of the API client library. However, upgrading is not necessary if you do not care about new features or bug fixes.

In addition, the YouTube app is delivered through the Google Play Store, which means that updates to the API service are not dependent on carrier or OEM system image updates. Generally, devices running Android 2.2 (Froyo) or later that have the Google Play Store app installed will receive updates within a few days. As a result, your application can use the YouTube Android Player API and reach most devices in the Android ecosystem.

Note: Users need to run version 4.2.16 of the mobile YouTube app (or higher) to use the API.

Getting Started.

The following documents will help you to set up your development environment and use the YouTube Android Player API:

The download page provides a link to download the API client library and JavaDocs.

The instructions for registering your application explain how to register your app in the Google API Console and to obtain an Android API key, which you will need to use the API.

The setup instructions explain how to set up an API project using either Eclipse or IntelliJ.

The sample applications overview describes the sample applications included in the API download.

The JavaDoc reference provides detailed definitions of the API's interfaces, classes, methods, and enums.

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 License, and code samples are licensed under the Apache 2.0 License. For details, see the Google Developers Site Policies. Java is a registered trademark of Oracle and/or its affiliates.

Android DownloadManager Progress.

I'm developing an app where users are able to download different content packages. For the download process I am using the DownloadManager class. That's working fine so far.

My question is how can I get the current progress of a running Download which was started with the DownloadManager. I know that there is the

buildin Download Notification and so on. But for me it is necessary that i get the progress of the running download so i can use it to show the progress in a custom progressbar in my app. So far i was not able to retrieve the progress.

Is it even possible or am i just blind and can't find the solution.

Hopefully someone can help me.

2 Answers 2.

I am looking for a better way of doing this also, but so far I am planning to just poll for progress every 1sec or so.

Edit:

A FileObserver can help with this. This is the skeleton of one I have put together to help keep track of which files our app has downloaded. Start it in an activity or service's onStart and stop it in onStop . Combined with a manual synchronization of the state of things during onStart , this can give you a pretty complete picture of what's going on.

For progress in particular, watching for the OPEN/CLOSE_WRITE events can help you decide when to start/stop polling the DownloadManager for updates.

How to Download File Using DownloadManager in API 29 or Android Q?

As I am new in Android Development, I am trying to simple App using DownloadManager.

Here is the code.

It works well when i run it on any device below API 29(My testing device was Nexus 5X ,Api 28 emulator). But when I run it on Nexus 5X ,API 29 The app gets crashed. Here is the Logs:

```
2019-09-24 20:51:46.355 15023-15023/com.blz.prisoner.downloadmanager D/AndroidRuntime: Shutting down VM.
```

I think the problem is on line "request.setDestinationInExternalPublicDir(getExternalFilesDir(Environment.DIRECTORY_PICTURES) + "/NewFile","sample2.jpg");" in DownloadImage(Uri uri) function. How to solve the problem??

Another Problem is when I run the app on the device below API 29 it runs well but when I click on notification after completing the download it doesn't open the image on gallery/on the folder it was saved.

6 Answers 6.

I solved just by using:

My relative path is:

I also have in my manifest:

To use Legacy storage management (Shared Storage) instead of new storage management (Scoped Storage) used from Android 10 and above.

Remember that by using "setDestinationInExternalFilesDir" files will be download to the external memory dedicated to your app, so: "external/Android/data/your_app_name/path_you_used_on_function". If you want to download it to another place you need to move It after you downloaded It by using Input & Output streams. To open the file with another app in Android version 10 or above you must use FileProvider.

If someone need it, this is the code to move (move, not copy. So the original file will be deleted. Remove "source.delete();" if you want to copy the file and not delete the source file) a file from one location to another:

Usage ("source" is the File you need to move, "path" is the destination):

Broadcast receiver for when the DownloadManager has finished:

Register it to the DownloadManager instance:

checkMakeDirs (this check if the dir exists or if it can make it successfully) and makeDirs (just make it without checking) code:

Important: from 05/07/2021 if your app is in the Google Play Store, theoretically, you must targetSdk=30 and also you must use Scoped Storage only to access your files (so use only the app-specific directory for your app). This means you need to use: