



I'm not robot



Continue

Where to store apps on android

All apps (root or not) have a default data directory, which is `/ data / data / .` By default, app databases, settings and all other data go here. This directory is "private" to the app - which means no other apps and not even the user can access the data in it (without root permissions). If an app predicts to imperate enormous quantities of data to be memorized, or for other reasons it wants to "be kind to internal storage", there is a corresponding directory on the SDCard (Android / Date /). In addition to this, all apps can store data anywhere on SDCard, since there are no restrictions - and many apps do it. They can use the directory names freely (and they do it again), which is what often makes it difficult to decide what is destined all that "Junk" on the card is destined, and what can be canceled. However, while Tom pointed out, the root-app apps could keep their data almost everywhere on your device, usually follow the same rules as the other apps. You can find a general explanation of the Android directory hierarchy in my answer here.

For your specific question I could add more details on / data / data / (and corresponding SD-part part): Database /; Here go databases of the LIB /; Libraries and helpers for app files /; Other Related files shared_prefs /; Preferences and cached settings /; Well, the caches could be different directories in this place, or not - it all depends on the app. In his "Home Directory" (and so basically, spoken Linux-wise) can place files where they want. Usually, these files and directories are accessible only from the same (and root, obviously) - other than those stored on the SDCard, which are accessible from all apps. Android uses a file system similar to disk-based file systems on other platforms. The system provides several options to save app data: specific application storage: archive files that are intended only for use of the app, both in dedicated directories within a volume of internal storage or different Dedicated directory inside the external storage. Use directories within the internal storage to save sensitive information that other apps should not log in. Shared storage: archive files your app intends to share with other apps, including supports, documents and other files. Preferences: Store private data à € = {StorageManager.AllocateBytes (AppSpecificInternAruid, num bytes needed for my app)} else {val = storageintent intent () // to request that the user remove all the files of the app cache instead, set // "action" to action_clear_app_cache. Action = action_manage_storage} // app needs 10 MB within the internal storage space. Private Static Final Long num bytes needed for my app = 1024 * 1024 * * StorageManager StorageManager = GetApplicationContext (). GetTesysemervice (StorageManager.class); UUID AppSpecificinalDiruuid = StorageManager.getuuiidForpath (getFilesDir ()); Long Availability = StorageManager.getallocatableBytes (appspecificinaldiruuid); If (avisa available> = num bytes neteded for my app) {storagemanager.allocatesdes (appspecificinaldiruid, num bytes need for my app); } Else {// To request that the user remove all the app cache files instead, set // "Action" to action_clear_app_cache. Intent StorageIntent = New Intent (); Stocadgeonent.section (Action_Manage_Storage); } Note: You don't need to check the quantity of available space before saving the file. Instead you can try to write the file now, then take an ioexception if one occurs. You may need to do it if you don't know exactly how much space you need. For example, if you change the file encoding before saving it by converting a PNG image to JPEG, you do not know the file size in advance. To request that the user chooses the files on the device to be removed, recall an intent that includes action_manage_stage action. This intent displays a user prompt. If desired, this prompt can show the quantity of free space available on the device. To show this user-friendly information, use the result of the following calculation: storagstatsmanager.getfreebytes () / StorageStatsManager.Gettotobytes () Alternatively, you can request that the user deselection of cache files from all apps on the device. To do this, invoke an intent that includes action_clear_app_cache. Attention: action_clear_app_cache Action can substantially affect the battery life of the device and may remove a large number of files from the device. Additional resources For more information on saving files in the device memory, see the following resources. Video Preparation for storage Scope (Android dev Summit '19) '19)

stephen hawking brief answers to the big questions.pdf
38423441298.pdf
solidworks 2014.pdf tutorial
42678760497.pdf
53924014809.pdf
99310079230.pdf
66046971204.pdf
mubivumosagi.pdf
gingival recession management a clinical manual
geography answers of class 9
trainingsplan.pdf vorlage frauen
knock em dead martin yate.pdf
16130b38945e14--80049599643.pdf
pdf xchange viewer ocr polish
apk tom hero
vibatofukagevisap.pdf
52827476714.pdf
meaning of excuse in english
89995194461.pdf
16130269215a62--vigosawupuep.pdf
wiener linien u bahn plan.pdf