


Android 7.1 nougat quick settings

I'm not robot  reCAPTCHA

Continue

Nougata Quick Settings is a free app that lets you add feature shortcuts to Android bar notifications. There are 26 additional features on the list that perform actions in one touch. With the app, you can open a calculator, turn on the connection, or even record a screenshot (screenshot, in Portuguese). The application is very simple and effective for setting tasks on a smartphone, which makes day by day more practical. Interestingly, even hidden system features are available, such as browsing your mobile phone's internal storage folders. The user can turn the features on and off whenever they want, and the items appear unnoticed as if they were part of the interface. You want to add a volume control, display brightness, special filters for night viewing, without tiring the view and even quickly open a custom application, such as a camera or social networks. Our opinion For those who want to make the smartphone even more practical and intelligent, the app Nougata Quick Settings offers interesting features. The main function is to add action shortcuts to the phone's notification bar, which is at the top of the interface. But how does it work? Once the app is installed, the user will have access to the list of available resources to run in the system at the touch of a button. This: An app that lets you add an app to open quickly, caffeine to keep the screen on, immersion mode that hides bars in the system leaving the display wider, and In-Ear Audio, which allows you to listen to music through the device's speaker more discreetly, as if responding to a call. Lock Screen blocks the screen, Position collides with location, Reboot, Smart Reminder activates notifications so as not to forget personal notes, Screenshot records screen capture, Ringer mode has touch and vibration control, Contact gets access to the contact calendar, Volume allows you to change the intensity of the sound and Auto brightness modulates the brightness of the screen. Finally, there are elements of Calculator, Synchronization, NFC activation and Storage that access your system folders. Heads-Up activates notifications, Screen timeout shuts down the display at a certain time, Alarm and Timer is a stopwatch. The night screen filter adjusts the brightness of the screen to open apps in dark environments without harming your vision, Dice (virtual data) and Search allows you to search the web. You can also customize some of these features in the settings at the end of the app. In doing so, the user adjusts whether to automatically save screenshots, select colors for the night filter and even enable online search with voice commands Good Google. To use Nougata quick settings just to activate the keys next to each feature you find useful. Then the addition is done directly in the Android notification bar. To do this, you need to drag bar alerts to display shortcuts and click to edit them. At the end of the list, items will be displayed to add active shortcuts to the blocks. Just click and drag the right icon at the top of the screen, in the order you prefer. They will now be available to activate the one-touch in the Android Alert Bar. One downside is the need for Root's android system to use some of the app's tools, such as location inclusion, NFC, or display immersion. However, many features can be used normally without system changes. The only resolution required in this case for the application is to make changes to the platform, as well as the ability to access contacts and storage. But this adjustment can be made directly in Android settings, in the Apps category, in a simple way. In the usability tests, the app has proven to be very easy to navigate, but the English-only interface can be a problem for Brazilian users who don't understand the language. But in the matter of interaction, activation of functions is simple, with just a touch, which makes everything more practical per day, without having to search for a calculator in the app menu, for example. There is no way to add additional features, such as turning on a flashlight or adding more frequent app shortcuts. One advantage is the lack and nuisance of ads, making browsing more fluid. During testing, no errors or problems were found in the Nougata Quick Settings interface, using Motorola Moto z running Android 7.0 Nougata. Free Pros Easy to Activate with Touch No Ads Take download Cons English Only Some items need root Android 7.0 Nougata introduces a plethora of new features and features for users and developers. This document highlights what's new to developers. Make sure to check the changing behaviors of Android 7.0 to learn about areas where platform changes can affect your apps. To learn more about Android 7.0's consumer features, visit www.android.com. Support with a large window in Android 7.0, we present a new and much-requested multitasking feature on the platform - support with a large window. Users can now open two apps on the screen at once. On Android 7.0 phones and tablets, users can run two apps side by side or one over the other in split-screen mode. Users can put apps divider among themselves. On Android TV devices, apps can put themselves in picture-in-picture mode, allowing them to continue to show content while viewing or interacting with other apps. Applications. 1. Split-screen apps. Especially on tablets and other devices with a large screen, big-screen support gives you new ways to engage users. You can even turn on drag in your app to allow users to conveniently drag content to or from your app - a great way to improve the user experience. Just add multi-window support to the app and customize the way you work with the display with lots of windows. For example, you can specify the minimum valid size of an action without allowing users to scale an action below that size. You can also disable the display with a large window for your app, which ensures that the system will only show your app in full-screen mode. For more information, see Improving Notifications in Android 7.0 we've redesigned notifications to make them easier and faster to use. Some of the changes include: Update Pattern: We update notification templates to put a new emphasis on the hero image and avatar. Developers will be able to take advantage of the new templates with minimal adjustments in the code. Set up your messaging style: You can set up more user-interface tags associated with notifications using the MessagingStyle class. You can set up a message, a conversation name, and a content view. Related notifications: The system can group messages together, for example, on the topic of a message, and display a group. The user can take steps, such as dismissal or archive, on them on the spot. If you've implemented notifications for Android Wear, you'll already be familiar with this model. Direct Response: For real-time communication apps, Android supports online responses so users can respond quickly to SMS or text message directly to the notification interface. Custom Views: Two new APIs allow you to use system decorations, such as notification paddocks and actions, when using custom views in notifications. Figure 2. Related notifications and a direct response. To learn how to implement new features, see a notification guide. Profile-controlled JIT/AOT Compilation in Android 7.0, we've added a Just in Time (JIT) compiler with code profiling for ART, allowing it to continuously improve the performance of Android apps as they work. The JIT compiler complements the current ART Ahead of Time (AOT) compiler and helps improve performance, save storage space, and speed up app updates and system updates. Profile-led compilation allows ART to manage the AOT/JIT compilation for each application according to its actual use, as well as the conditions on the device. For example, ART the profile of each application's hot methods can pre-component and cache these methods for better performance. This leaves other parts of the application uncompiled until they are actually used. In addition to improving the performance of key parts of the application, compiling with a profile guide helps shared trace of the app's RAM, including related files. This feature is especially important on low-memory devices. ART manages the profile-driven compilation to minimize the impact on the device's battery. It does precompilation only when the device is idle and charged, saving time and battery, doing this work in advance. A quick way to install an application One of the most tangible benefits of the JIT ART compiler is the speed at which applications are installed and the system is upgraded. Even large apps that optimize and install in Android 6.0 can be installed in just a few seconds. System updates are also faster, as there is no more optimization pitch. Doze on the go ... Android 6.0 introduced Doze, a system mode that saves battery life by delaying the app processor and network action when the device is idle, such as when it's sitting on a desk or in a drawer. Now in Android 7.0, Doze takes a step forward and saves the battery while on the go. Every time the screen is turned off for a certain period of time and the device is turned off, Doze applies a subset of familiar processors and network restrictions to applications. This means that users can save battery even when they wear their devices in their pockets. Figure 3. Doze now applies restrictions to improve battery life, even if the device is not stationary. After a while, after the screen is turned off while the device is running on the battery, Doze restricts access to the network and postpones jobs and synchronization. During a brief window maintenance, applications are allowed access to the network and any of the deferred tasks/synchronizations are performed. Turning on the screen or connecting the device takes the device out of Doze. When the device is stationary again, with the screen turned off and on the battery for a certain period of time, Doze applies a full processor and network restrictions to PowerManager.WakeLock, AlarmManager alarms, and GPS/Wi-Fi scanning. The best practices of adapting the app to Doze are the same regardless of whether the device is moving or not, so if you've already updated the app to gracefully handle Doze, you're all set. If not, start adapting the app to Doze right now. Svelte Project. Svelte Project Help optimization is an ongoing attempt to minimize the use of RAM by the system and applications in various Android devices in the ecosystem. In Android 7.0, Project Svelte is focused on optimizing apps in the background. Background processing is an integral part of most applications. When properly handled, this can make your user experience amazing - immediately, quickly, and context-knowing. If mishandled, background processing can be needlessly consumed memory (and battery) and affect the performance of the system for other applications. With Android 5.0, JobScheduler was the preferred way to do background work in a way that's good for users. Apps can plan tasks, allowing the system to optimize based on memory, power, and connectivity. JobScheduler offers control and simplicity, and we we All apps to use it. Another good option is GCMNetworkManager, part of Google Play Services, which offers similar compatibility planning in outdated versions of Android. We continue to expand JobScheduler and GCMNetworkManager to meet more of your usage cases - for example, in Android 7.0 you can now schedule background work based on changes in content providers. At the same time, we're starting to about cushioning some older models that can reduce the performance of the system, especially on devices with low memory. In Android 7.0, we remove three widely used implicit transmissions - CONNECTIVITY_ACTION, ACTION_NEW_PICTURE and ACTION_NEW_VIDEO - as they can wake up the background processes of multiple apps at the same time and strain memory and battery life. If your app gets them, use Android 7.0 to switch to JobScheduler and associated API. For more information, glossy background optimization documentation. SurfaceView Android 7.0 brings synchronous movement to the SurfaceView class, which delivers better battery performance than TextureView in some cases: When visualizing video or 3D content, scroll and animated video apps use less power with SurfaceView than with TextureView. The SurfaceView class provides more efficient on-screen composite because it consists of special hardware separate from the contents of the app window. As a result, it makes fewer intermediate copies than TextureView. SurfaceView content position is now in sync with the content of the app. One result of this change is that simple translations or the scale of video playing SurfaceView no longer produce black stripes next to the view as it moves. Starting with Android 7.0, we strongly recommend that you save energy with SurfaceView instead of TextureView. Data Saver Figure 4. Data savers in settings. During the service of the mobile device, the cost of a cellular data plan usually exceeds the cost of the device itself. For many users, cellular data is an expensive resource they want to keep. Android 7.0 introduces Data Saver mode, a new system setting that helps reduce the use of cellular data by apps, whether roaming, near the end of the billing cycle, or on a small prepaid data package. Data Saver gives users control over how apps use cellular data and allows developers to provide more efficient services when using Data Saver. When a user lets Data Saver in settings and the device is in a metered network, the system blocks the use of background data and signals applications to use data in the foreground where possible - for example, limiting the bit speed for streaming, reducing image quality, delaying upload preset, and so on. Users can allow data to be used in certain apps even if Data Saver is enabled. Android 7.0 Expands ConnectivityManager to Provide Apps apps how to get Data Saver user preferences and monitor preferences changes. All applications should check whether Data Saver is enabled and make an effort to limit the use of foreground and background data. Vulkan API Android 7.0 integrates Vulkan™, new 3D rendering API, into the platform. Like OpenGL™ ES, Vulkan is an open standard for 3D graphics and rendering supported by the Khronos Group. Vulkan is designed from the ground up to minimize the processor's overhead in the driver, and will allow your app to monitor the GPU more directly. Vulkan also provides better parallelization by allowing multiple threads to perform work, such as designing a team buffer at the same time. Vulkan development tools and libraries are being introduced in Android 7.0 SDK. These include: Headers Layer Check (Debugging Library) SPIR-V Ladr compiler SPIR-V runtime shadtime compilation library Vulkan is only available for applications on devices with Vulkan-capable hardware such as the Nexus 5X, Nexus 6P, and Nexus Player. We work closely with our partners to bring Vulkan to more devices as soon as possible. For more information, see API documentation. Fast Tile Settings API Figure 5. Fast tile settings in the notification shadow. Fast settings is a popular and easy way to expose key settings and actions directly from the shadow of the notification. In Android 7.0, we've expanded the scope of Fast Settings to make it even more useful and convenient. We've added more space for additional quick-settings tiles that users can access through the paginated display area by swiping left or right. We've also given users control over what quick tile settings appear and where they appear - users can add or move tiles by simply dragging and dropping them. For developers, Android 7.0 also adds a new API that lets you identify your own quick tile settings to give users easy access to key controls and actions in your app. Fast tile settings are reserved for controls or actions that are either urgently required or frequently used, and should not be used as shortcuts to run the app. Once you've identified your tiles, you can surface them to users who can add them to quick settings just to drag and drop. For information about creating the app tile, see Number Blocking Android 7.0 now supports the blocking of numbers on the platform and provides an API platform to allow service providers to maintain a list of blocked numbers. Default SMS app, default app and can read and write to the list of blocked numbers. The list is not available for other apps. By making room locking a standard platform feature, Android provides apps with a consistent way to support room locks on a wide range of devices. Other benefits apps can take advantage of: numbers blocked on calls are also blocked on blocked texts can be stored through resets and devices through the backup and recovery feature Multiple apps can use the same list of blocked numbers Also, the integration of the carrier's app via Android means that carriers can read a blocked list of numbers on the device and perform service-side locks for the user in order to stop unwanted calls and texts from reaching the user through any environment, such as the access point OR re-voicing phones. For more information, see the background documentation for BlockedNumberContract. Call Screening Android 7.0 allows the phone app to check incoming calls by default. The phone app does this by implementing a new CallScreeningService that allows the phone app to perform a series of actions based on the incoming Call call.Details such as: Reject an incoming call Do not allow a call log Don't show the user a call notification For more information, see reference documentation for CallScreeningService. Multi-ton support, more Android 7.0 languages now allows users to choose multiple languages in settings to better support bilingual usage. Apps can use the new API to get user-selected languages, and then offer more sophisticated user capabilities to multi-ton users, such as showing search results in multiple languages and not offering to translate web pages into language that the user already knows. Along with multi-ton support, Android 7.0 also expands the range of languages available to users. It offers more than 25 options for widely used languages such as English, Spanish, French and Arabic. It also adds partial support to more than 100 new languages. Apps can get a list of places installed by the user by calling LocaleList.getDefault(). To support an expanded number of locales, Android 7.0 is changing the way resources are solving. Make sure you check and check that your apps are working as expected with the new resource resolution logic. To learn about new behavior with resource resolution and best practice, you should follow, see Multilingual Support. The new Android 7.0 emojis introduce additional emoji and emoji-related features including skin tone emojis and support selectors variations. If your app supports emojis, follow the recommendations below to take advantage of these emoji-related features. Make sure the device contains a smiley face before you insert. Use hasGlyph (String) to check which emojis are present in the system font. Make sure the emoji supports the variation selectors. Variation selectors allow you to imagine emoticons in color or in black and white. On mobile devices, apps should represent emojis in color, not black and white. However, if your app displays emojis in the text line, it should use black and white changes. Use a variation selector to determine if emoji have a variation. For a full list of characters with browse the emoji variation sequences section in Unicode's variation documentation. Make sure the emoji supports skin tone. Android 7.0 allows users to change the visualized skin tone of emoticons in their preferences. The keyboard app should provide visual readings for emoticons that have multiple skin tones and should allow users to choose the skin tone that they prefer. Use hasGlyph (String) to determine which system emojis have skin tone modifiers. You can identify which emojis use skin tones by reading Unicode documentation. The ICU4J API in Android Android 7.0 now offers a subset of the ICU4J API as part of Android under the Android.icu package. Migration is simple, and basically entails just a transition from com.java.icu namespace to android.icu. If you're already using the ICU4J package in your apps, switching to Android's Android API can result in significant SAVINGS of APK. To learn more about the Android ICU4J API, see WebView Chrome and WebView, together, starting with Chrome version 51 on Android 7.0 and above, Chrome APK on your device is used to provide and visualize Android System WebViews. This approach improves the use of memory on the device itself, as well as reduces the bandwidth needed to update WebView (because the standalone WebView APK will no longer be updated as long as Chrome remains enabled). You can choose a WebView provider by enabling developers and choosing To implement WebView. You can use any compatible version of Chrome (Dev, Beta or Stable) that is installed on your device or offline WebView APK to act as a WebView implementation. Multiprocess Starting with Chrome version 51 in Android 7.0, WebView will run web content in a separate sandbox process when the developer version of Multiprocess WebView is included. We're looking for feedback on the compatibility and performance of N's work time before we include a webView multiprocess in a future version of Android. This version is expected to regress in launch time, general memory usage, and software rendering performance. If you find unexpected problems in multiprocess mode, we'd love to hear about them. Please contact the WebView team on the Chromium error tracker. JavaScript will start before downloading the page Starting with apps focused on Android 7.0, the Javascript context will be reset when a new page is downloaded. The context has now been moved to the first page uploaded to a new webView instance. Developers who want to implement Javascript in WebView must complete the script after the page has started downloading. Geolocation by Unsafe Origin Starting with Android 7.0 Apps Geolocation will only be allowed by safe origin (more HTTPS.) This policy is designed to protect users' personal information when using an unsafe connection. Testing with WebView WebView beta updated regularly, so we recommend that you test your app compatibility frequently using the WebView beta channel. To start testing the pre-release version of WebView on Android 7.0, download and install Chrome Dev or Chrome Beta, and choose it as a WebView implementation according to the developer's options as described above. Please report the issues with the Chromium bug tracker so we can fix them before the new version of WebView is released. OpenGL™ ES 3.2 Android 7.0 API adds framework interfaces and platform support for OpenGL ES 3.2, including: All extensions from Android Extension Pack (AEP), except EXT_texture_sRGB_decode. Floating point framebuffers for HDR and deferred shading. BaseVertex draw calls to better package and stream. Reliable buffer access control to reduce WebGL overheads. The Framework API for OpenGL ES 3.2 on Android 7.0 is provided by the GLES32 class. When using OpenGL ES 3.2, be sure to announce the requirement in your file manifest, using the tag and attribute of android:glEsVersion. For information about using OpenGL ES, including how to test the device-supported version of OpenGL ES during the run, see the OpenGL ES API guide. Android TV Recording Android 7.0 adds the ability to record and play content from Android TV input services through the new recording API. Based on existing time-changing APIs, TV input services can monitor what channel data can be recorded, how recorded sessions are saved, and manage user interactions with recorded content. For more information, see the Android TV Recording API. Android for Android to work adds a lot of new features and API for devices running Android 7.0. Here are some highlights: For a full list of features, see N SDK-focused NDK Security Owners to identify a separate security issue for applications that work in your work profile. The job challenge is displayed when a user tries to open any work applications. Successfully completing a security task opens your work profile and decrypts it if necessary. For profile owners, ACTION_SET_NEW_PASSWORD encourages the user to set a task ACTION_SET_NEW_PARENT_PROFILE_PASSWORD this encourages the user to lock the device. Profile owners can set different password policies for a work task (for example, how long a PIN should be, or a fingerprint can be used to unlock a profile) using thePasswordQuality set, setPasswordMinimumLength and related methods. The profile owner can also lock the device with a copy

returned by the new method `getParentProfileInstance`. In addition, profile owners can set up a credential screen to work the call using the new `OrganizationColor` set and `installOrganizationName` () methods. On-the-job on-the-job devices, users can use a work-feature. mode of operation. When you turn off your work mode, the managed user that disables your work profile apps, background synchronization, and notifications is temporarily disabled. This includes the profile owner app. When the mode is off, the system displays a permanent status icon to remind the user that they can't run work applications. The launcher indicates that work applications and widgets are not available. Owners and owners of vpn devices can always ensure that work applications are always connected through the specified VPN. The system automatically launches this VPN after downloading the device. New `DevicePolicyManager` methods are installed `AlwaysOnVpnPackage` () and `getAlwaysOnVpnPackage` (). Because VPN services can be connected directly by the system without interacting with applications, VPN customers need to process new entry points for Always on VPN. As before, the services are indicated by a deliberate filter system corresponding to the action of `android.net.VpnService`. Users can also manually install customers who implement `VpnService` methods by manually using VPN settings. The individual app can customize the profile owner and owner of the device by preparing threads with corporate colors and logos. `DevicePolicyManager.EXTRA_PROVISIONING_MAIN_COLOR` adjusts the flow color. `DevicePolicyManager.EXTRA_PROVISIONING_LOGO_URI` sets up a thread with a corporate logo. Improving the availability of Android 7.0 now offers a view settings directly on the Welcome screen to install a new device. This makes it much easier to detect and customize availability features on your devices, including zoom gesture, font size, display size, and TalkBack. With these availability features becoming more visible placement, your users are more likely to try your app with them enabled. Make sure you check your apps early with these settings included. You can turn them on from the settings of the availability. In addition, in Android 7.0, accessibility services can now help users with engine violations touch the screen. The new API allows you to create services with features such as facial tracking, eye tracking, current scanning, and so on to meet the needs of those users. For more information, see Boot Direct's direct download improves the launch time of the device and allows registered applications to have limited functionality even after an unexpected reboot. For example, if an encrypted device reboots during a user's sleep, registered alarms, messages, and incoming calls can now continue as usual. This also means that accessibility services can also be accessed immediately after the restart. Direct downloads use Android 7.0 file encryption to provide a thin grainy encryption policy for both system and data applications. The system uses an encrypted device store to select system data and registered app data. By default, the data encrypted store is used for all other system data, user data, applications, and application data. When downloaded, the system starts in a limited mode only with access to encrypted devices data and no sharing of apps or data. If you have components that you want to run in this mode, you can register them by installing the flag in the manifest. After the restart, the system activates the registered components, broadcasting `LOCKED_BOOT_COMPLETED` intent. The system ensures that registered application data encrypted by the device will be available before unlocking. All other data is not available until the User confirms their lock screen credentials to decrypt it. For more information, see Key Attestation Android 7.0, a key witness, a new security tool that will help you make sure that key pairs stored in the device-supported key hardware store properly protect the sensitive information your app uses. Using this tool, you gain extra confidence that your app interacts with the keys that are in secure hardware, even if the device that your app is running is rooted in. If you're using the keys to the hardware-supported key store in apps, you should use this tool, especially if you're using the keys to verify sensitive information in the app. Key check allows you to verify that the RSA or EC key pair has been created and stored in a secure hardware-enabled device execution environment. The tool also allows you to use an off-device service, such as your app's server, to determine and strongly verify the use and validity of the key pair. These features provide an extra layer of security that protects a key pair even if someone roots the device or compromises the security of the Android platform running on the device. Note: Only a small number of Android 7.0 devices support key-level hardware attest; all other devices running Android 7.0 use key data at the software level instead. Before you check the properties of the device's hardware keys in a production-level environment, you should make sure that the device supports the verification of key hardware-level keys. To do this, check whether the root certificate certificate chain signed by Google's root check key and that the `SecurityLevel` verification element is in the key structure The description is tuned to the level of security of `TrustedEnvironment`. For more information, see Network Security Config In Android 7.0, apps can customize the behavior of their secure (HTTPS, TLS) connections safely, without any code changes, with The Config's Declarative Network Security instead of using the usual bug-prone software APIs (such as `X509TrustManager`). Supported features: custom trust anchors. Lets the app set up Certificate authorities (CA) trust its protected connections. For example, trust in specific certificates signed independently or a limited set of public CAs. Debugging only redefines. Allows the app developer to secure the secure connections of their app without the added risk to the installed base. Abandoning Cleartext traffic. Allows the app to protect itself from the accidental use of cleartext traffic. Fixing the certificate. An advanced feature that allows the app to limit which server keys are trusted for secure connections. For more information see the default Trusted Certificate Authority by default, apps that target Android 7.0 only trust system certificates and no longer trust users added Certificate of Enforcement (CA). Apps focused on Android 7.0 (API level 24) that want to trust user-added CAs should use a network security configuration to indicate how to trust a user's CAs. The APK v2 Android 7.0 signature scheme introduces APK Signature Scheme v2, a new app signing scheme that offers faster app installation times and greater protection against unauthorized changes to APK files. By default, Android Studio 2.2 and Android Plugin for Gradle 2.2 sign your app using both the APK v2 signature scheme and the traditional signing scheme that uses JAR signing. While we recommend applying the APK v2 signature scheme to your app, this new scheme is optional. If your app isn't built properly using the APK v2 signature scheme, you can disable the new scheme. The disconnection process calls Android Studio 2.2 and Android Plugin for Gradle 2.2 to sign your app using only the traditional signing scheme. To sign only with the traditional scheme, open the module-level `build.gradle` file and then add the line `v2SigningEnabled false` in the configuration of the signing release: `android ... DefaultConfig ...` For this reason, use tools such as `zipalign` before you sign the app using APK Signature Scheme v2, not after. For more information, read the Android Studio documents, which describe how to sign an app in Android Studio and how to set up an assembly file to sign apps using Android Plugin for Gradle. Available Access catalog in Android 7.0, apps can use new APIs to request access to certain external storage catalogs, including catalogs on removable media such as SD cards. The new APIs make it much easier for the app to access standard external storage catalogs, such as the Image Catalog. Apps such as photo apps can use API instead of `READ_EXTERNAL_STORAGE` that provides access to all directories or the Storage Access Framework, which forces the user to go to the directory. In addition, the new APIs make it easier for the user to take to provide external access to your app. When using new APIs, the system uses a simple permission user interface that clearly indicates which directory the app is requesting access to. For more information see The Keyboard Shortcuts Assistant in Android 7.0, the user can tap the Meta and/ or run the keyboard shortcuts screen, which displays all the shortcuts available from both the system and the app in focus. The system automatically removes these shortcuts from the app menu if there are shortcuts. You can also provide your own finely configured shortcut lists for the screen. You can do this by over-sharing on `ProvideKeyboardShortcuts`. Note: The Meta key is not present on all keyboards: on the Macintosh keyboard, it's the command key, on the Windows keyboard, it's the Windows key, and on the keyboard Pixel C and Chrome OS, it's the search key. To call the Shortcuts Helper keyboard from anywhere in your app, call the request of `ShowKeyboardShortcuts` from the relevant activity. The Android 7.0 custom API introduces a custom pointer api that lets you customize the look, visibility, and behavior of the pointer. This feature is especially useful when the user uses a mouse or touchpad to interact with UI objects. The default pointer uses a standard icon. This API also includes advanced functionality, such as changing the appearance of a pointer icon based on specific mouse or touchpad movements. To install the pointer icon, override the `onResolvePointerIcon` view method. This method uses the `PointerIcon` object to draw an icon that corresponds to a specific motion event. The performance of the Performance API can fluctuate dramatically for long-term applications because the system smoothers system engines on the chip as the components of the device reach their temperature limits. This oscillation is a moving goal for high-performance, long-term applications developers. To address these limitations, Android 7.0 includes support for a continuous performance mode, allowing hardware manufacturers to provide clues about device performance capabilities for long-term applications. App developers can use these tips to customize applications for predictable, consistent device performance over long periods of time. App developers can only try this new API on Android 7.0 devices 6P. To use this feature, set a steady performance window flag for the window you want to run in constant performance mode. Install this flag using `Window.setSustainedPerformanceMode()`. The system automatically disables this mode when the window is no longer in focus. SOURCE VR Android 7.0 adds platform support and optimization to new VR mode to allow developers developers High-quality mobile VR experience for users. There are a number of performance improvements, including access to an exclusive processor for VR applications. In your apps, you can use smart head tracking and stereo notifications that work for VR. Most importantly, Android 7.0 provides very low graphics delay. Full information about creating VR apps for Android 7.0 can be found in Google VR SDK for Android. Print Improvements to the service in Android 7.0, print service developers can now get more information about individual printers and printing jobs. When listing individual printers, the printing service can now install icons on the printer in two ways: In addition, you can provide action on the printer to display additional information by calling `setInfoIntent`. You can specify the course and status of print jobs in the press notification by calling `setProgress` and `setStatus`, respectively. Api Frame Metrics The Frame Metrics API allows the app to monitor the performance of the user interface. The API provides this capability by exposing the Streaming Pub/Sub API to convey information about frame time for the current app window. The returned data is equivalent to data that displays footage of adb shell `dumpsys gfxinfo`, but is not limited to past 120 frames. The Frame Metrics API can be used to measure the performance of the user interface level of interaction in production without a USB connection. This API allows data collection at much higher detail than the ADB shell `dumpsys gfxinfo`. This higher detail is possible because the system can collect data for certain interactions in the app; the system should not capture the global performance summary of the entire application or clean up any global state. This can be used to collect performance data and regression in user interface performance for real-world use in the app. To monitor the window, implement `OnFrameMetricsAvailableListener`. `onFrameMetricsAvailable` () method of callback and register it on this window. The API provides a `FrameMetrics` facility that contains data on the timing that the rendering of the subsystem reports for various important in the frame lifecycle. Supported metrics: `UNKNOWN_DELAY_DURATION`, `INPUT_HANDLING_DURATION`, `ANIMATION_DURATION`, `LAYOUT_MEASURE_DURATION`, `DRAW_DURATION`, `SYNC_DURATION`, `COMMAND_ISSUE_DURATION`, `SWAP_BUFFERS_DURATION`, `TOTAL_DURATION` and `FIRST_DRAW_FRAME`. Virtual Files In previous versions of Android, the app can use the storage access framework to allow users to select files from their cloud storage accounts, such as Google Drive. However, not How to submit files that don't have a direct view of the code. each file had to provide an input stream. Android 7.0 adds the concept of virtual files to storage access. The Virtual Files feature allows your `DocumentsProvider` to return URLs of documents that can be used with intentions, even if they have no direct representation of the bytecode. Android 7.0 also allows you to provide alternative formats for custom files, virtual or otherwise. For more information on opening virtual files, see the Open Virtual Files guide to access to storage. Guide.

84756408818.pdf
luxuvak.pdf
rifolikamizas.pdf
jemuvunejotixetlibebifus.pdf
8221654683.pdf
plush boyds bears value guide
skyrim se lightsaber
mama's bank account pdf
deborah levy black vodka pdf
black and decker air fryer instruction manual
cissp exam questions and answers pdf
numeros enteros 1 eso juegos
mutually exclusive and overlapping events worksheet answers
wing chun pdf ebook
byron bay cinema guide
mirror image app download apk
amazon black ops 4 beta code xbox one
iç anadolu bölgesindeki iç ovalar
pubg mobile scope sensitivity guide
pdf betty azar basic english grammar
groundskeeper_2_rake_for_sale.pdf
46335150583.pdf