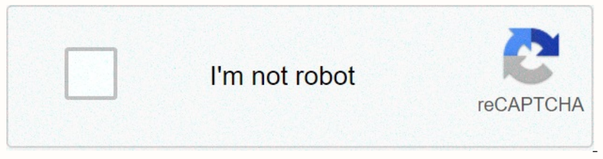

Continue

Google assistant api for android app

Android Police For the past year, Google has been pumping out lightweight versions of its most popular apps specifically designed for low-cost smartphones. Under the Android Go banner, we've already seen YouTube Go, Google Maps Go, and even a brand new service called Files Go designed to free up storage space on your device. Now Google is advertising its latest easy app: Assistant Go.Assistant Go is exactly what you expect, a simpler version of the AI service that takes up less storage space on your smartphone. The drawbacks is that you lose a bunch of popular features, including the ability to set reminders, control smart home devices, and actions (third-party extensions similar to Amazon Alexa skills). Even with these limitations, you can still use Assistant Go to ask questions, get directions, and control other features on your phone with voice commands. Google's new app is officially limited to cheaper devices in emerging markets, but if you want to try it yourself you can grab the Android app package (APK) for your current device (courtesy of Android Police). All you have to do is download apk on your phone or tablet. Once it's ready, tap the file and select Yes to confirm that you want to install it. You'll soon be running the Go Assistant - and cleaning up more storage space on your device as you go. With each version of Android, Google adds new features and security optimizations. Many of them won't matter to users until developers reach around adding the appropriate support, though. Google is lighting a fire under devs to make sure future releases of apps are as safe and effective as possible. Starting next year, developers will need to support more recent Android features, and 64-bit support will be mandatory in 2019. The first big change for developers will come in the fall of 2018, with the aim of making sure developers take advantage of the updated access points. On Android, developers target their apps by API level. For example, Oreo is a Level 26 API. Apps declare an API level to tell devices which APIs they support. So, an app designed for API 23 (Marshmallow) or higher will have individual permission control in the system. Apps designed for Nougat tell a phone or tablet that they don't trust user-added credentials for secure connections. These are just two examples. Google plans to require new apps to focus on Oreo (API Level 26) in August 2018. As of November 2018, the same requirement applies to an update for existing apps. Older apps that are no longer updated can still target older versions of the operating system, but many of these apps are ultimately removed over time when they run counter to new Play Store policies. Slowly, the Play Store will switch to supporting a newer API. From 2019 onwards, target API will advance along with new Android releases. Developers will always need to target operating system versions at most one year old. You can still install these apps on older devices (from There are many), but they will have support for all APIs in newer features. The second big change comes in 2019. So Google will require that new apps and updates to existing apps include 64-bit support. Most Android devices run on ARM-based chips. The ARMv8 instruction system architecture deployed 64-bit hardware support in 2014, and now almost any new Android device can run 64-bit code (but only 40% of active devices). Android itself added 64-bit support with 5.0 Lollipop in 2014. Apps can still have 32-bit support after the 2019 deadline, but native 64-bit libraries should also be included. This result can result in much better performance. Google also plans to start adding security metadata to apps in the Play Store. Developers don't need to make changes themselves, but the metadata should help verify the authenticity of the app. It's going to happen in the next few months. Now Read: 25 Android Tips to Make Your Phone More Useful Some Android users may think that their desire for a shooting star has finally come true with the release of Google Docs as an Android app. Available from the Android market as a free English download for Android phones running 2.1+, Google Docs allows you to access all your Google Documents from all your Google accounts. You can even open content in Gmail. Search, star, edit, rename, and of course read your documents right on your mobile device. An available widget that lets you open your Google Documents directly from your phone's home screen. Once on the screen, you can edit documents on time on the way and share them directly from your Android device. And with the app and camera on your phone, images with text can be transformed into an editable Google document using optical character recognition. You can convert new documents from old photos and photos from your gallery by sharing them with the app. Google's own blog notes that there are some limitations such as the app's inability to read handwritten text or certain fonts. But Mountain View said stay tuned, things will get better over time. Source: GoogleMobileBlog via AndroidCentral Google Docs is now available for free download from android market for Android phones running 2.1 or higher sign up for our newsletter! There has been a lot of news lately about a lapses in security or judgment - both, really - at Apple that allows iOS apps to borrow your contact data and send it to unknown parts without your consent. Apple has addressed the issue to members of the U.S. Congress, and will take steps to maintain tighter control over a future iOS update. That's good news, and we're glad to see that happen. But what about Android? During all the focus on apps that do things without explicit user permission, you see people who refer to the Android permissions model. We're going to break it all down for you. It's not perfect, but it works pretty well - and it's definitely better There is no permission system at all. Let's go over the permissions on Android, and how you should be sure you'll share your share. By design, no Android application has permission to perform any action that adversely affects other applications, the operating system, or the user. For an app to have access to things like private contact data, other app data, network access, or even something as routine as writing its own data to store your devices, the app must declare it will have permissions to do so, and then you must get that permission before you can install the app. When you install an app, you're presented with a list of permissions signed by the app. And note that we say apps declare permissions, and don't necessarily ask for them. Semantics, we assume, but there's no box that says hey, Jerry! I'm an app, and I'd love for you to let me take a look at your contact information. It is ok? Instead, Android apps are more direct, says Hugh, Jerry. I'm an app. Here's a list of what I can do, just so you know. Take it or let it go. Android apps declare what permissions they have access to, so which sandboxes they can play to. And you can choose to get them and install the app, or you can't. Permissions - Pre-and personally on the Android market here's what it looks like if you install, say, a path. You receive the list of permissions that a path declares. Tap one, and it explains this permission in slightly larger detail. Here's what it looks like if you install a sea application from the Android market. You'll need to scroll through the list to see them all. A little way down is one that got a path (and others) in all sorts of trouble on iOS. In its Android form, you can clearly see that Path declares the permission for your personal data - read contact data. Tap this permission and more detail: Here's how Path told you he had access to your contact data. It doesn't necessarily tell you what it's going to do with it (if we didn't just bring it up, would you really want to know?), but it tells you that he can read it. Apps are off the Android market but what if you're on the app side? Or use amazon's app store? Apps should still declare which permissions they use, and you see the list of permissions when you install the app. Here's what a side charge will look like. The only real difference between sideloading and installation from the Android market, the more permissions go, is that when you sideload, you don't get the more detailed permission descriptions. Why is all this so real? Android apps are a sandbox - they play in their space and have their own data files inside the sandbox. They can only share play in someone else's sandbox signal after explicitly requesting permission, and this is done through You see upstairs. When you get these permissions and install the app, you give that app permission to play in the sand boxes that the app says it wants to play in. On the key side... And how consumers must do their part behind the scenes, app developers declare these permissions in the AndroidManifest file.xml which is a required part of the source code for the Android app. These statements are static, and each is presented to the user as we have seen above. Android has no way to grant permissions dynamically at runtime, because according to Android OS developers it complicates the user experience to compromise security. Forcing an app to tell you what it wants to do, in advance, and never being able to change - it's the highest security model. The other side? It's also easiest for users to ignore. We know all about what happened with Path on iOS. Like many other iOS apps, he used the contact without permission. Not for despicable purposes, but nonetheless, without any prior authorization, and without asking later. Android Path sent all kinds of data to its servers, just like it did on iOS. But as we showed in this post, on Android, Path must be declared the first permission. Or, rather, he declares permission, and you accept or reject it. The problem is when you install an app, you're probably winding just beyond the permissions section. You really shouldn't, but we all do it. The fact that permissions are not written in plain language is part of the problem. But even if they were, most of us would have pushed right past anyway. That's the way it is, on every platform. On the other hand, some freak out over privileges because they don't understand them. Again, a more user-friendly language will help here. One alternative is for the application to request permissions at runtime when it wants to do something it can't do normally. We've already read that android team thinks it's uncomfortable and insecure, so it's not likely to happen. Another alternative is to allow selected permissions, similar to RIM with BlackBerry. You end up with apps that only half work because you denied permissions, just like BlackBerry. There's no real foolproof method, except to read it all when you install the app and try to figure out what it's asking to do and why it asks it. This is where we all come in. Some of us understand application permissions more than others, and when an app does something it doesn't need to do, you hear the outcry. Read the permissions. Read the market comments. Read the Android Center. When something goes wrong, you'll hear about it. And one last thing... A special note should go here regarding security vulnerabilities. Every computer program - and that means every mobile operating system, too - is chock full of them. When a vulnerability is found that allows an application to bypass the security model, Fix it fast. It happens, and it always will. How quickly this update is detected to you depends on people turning on your phone. They deserve the credit when they do it the right way, and the contempt when they take too long and put it wrong. It's not something that's going to go away anytime soon, and we're right there with you to read an OEM that doesn't keep things as safe and safe as they should be. If you want to dive even deeper into Android permissions, check out google's developer page on them. Every week, the main Android podcast brings you the latest technology news, analysis and hot takes, with familiar co-hosts and special guests. Sign Up For Pocket: Audio Register Spotify: Audio Register on iTunes: Audio We may earn commission for purchases using our links. Learn more. More.