How Google Smart Lock can improve user sign-up

Google Smart Lock

Quickly access your favorite apps and sites across Android and Chrome. Smart Lock helps you use difficult passwords without remembering or retyping them. Learn more

The best app login experience is one that isn't an "experience" at all. One that requires as little effort as possible from the user in order to get them to where they really want to go as quickly as possible.

Ideally, users should be able to access the bulk of your app's content without the need to login at all, but in situations where it is still a requirement, easing the pain as much as possible is a key factor in reducing the loss of users in this critical stage.

Nowadays the old username and password paradigm just doesn't cut it.

Remembering login credentials for every different service we use is just not feasible, which leads to many people using the same password repeatedly, or third party password managers such as LastPass and 1Password. We shouldn't have to explain the issue with the former approach, and the latter can be significantly more cumbersome on a mobile device. For these reasons, it's always important to consider the different login options available to us in mobile apps.

The one-click solution
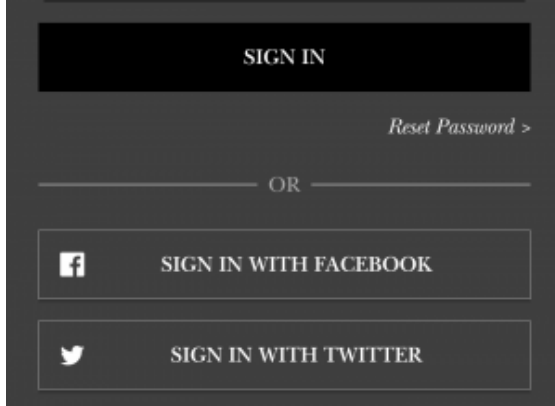
In the ever increasingly "social" world we live in today, there's a high possibility that a user of your app will already have an authenticated account with another popular service such as Facebook or Twitter. Knowing this, we can ease the password pain by offering users a one-click, "federated" login using one of these services. Instead of forcing users to create new login credentials for our app, we ask a third party to authenticate the user using their existing credentials. Since a lot of these users will already have the third party's app installed on their device, this often results in a one-click process.

For most users, this provides a quick, easy solution for logging into apps. There's no need to remember any new passwords or enter a lengthy email address, it just works. Federated logins also provide us with a quick and easy way to sign up **new** users, as sometimes it is possible to pull out key pieces of information from the user's existing social account and use this to fill in the new account for our app.

Although federated logins provide a much better experience, privacy concerns mean that some users still do not feel comfortable using them, and in some countries these services aren't even available in the same capacity we often take for granted, making these options not so "quick and easy".

Another downside to this solution is what is often referred to as the "NASCAR" login screen; a screen filled with third party logos that doesn't really fit in with the rest of the app's design.

Easing the password pain

For years internet browsers such as Chrome, Firefox and Safari have been offering the ability to store credentials for each website you login to, and in more recent years even synchronising these to all your devices. Until recently, this functionality has been sorely lacking on mobile devices, specifically when it comes to Android.

On the iOS side of things, Apple's solution to password hell is the **iCloud Keychain**. When enabled on each of your iOS and OS X devices, iCloud Keychain will keep passwords for Safari, accounts and Apple's own apps up to date, allowing easier logins across the board. The service is also available for third party apps and can be used on devices with iOS 7 and above. Despite being around for quite some time now, the majority of apps available on the App Store seem to have overlooked the value of this feature, possibly due to the need for users to manually enable it on each device.

On Android, Google's **Smart Lock for Passwords** goes a long way to solving the issues around password logins. Smart Lock differs from iCloud Keychain in that users do not need to manually enable the feature; instead it can offer the chance to store login details on a successful app login, making each individual login an opt-in, rather than the whole service. This should go some way to improve discoverability of the feature and hopefully prompt more developers to implement it in their apps.

On subsequent logins, users can be automatically logged in using the stored credentials from Smart Lock, or asked to choose a set of credentials if more than one exists. This makes any subsequent logins a one or even zero click process!

Credentials are also synchronised between Chrome and Android, meaning that if a user already opted to save their password when signing into your website in Chrome, logging in on their mobile device may be as simple as opening your app.
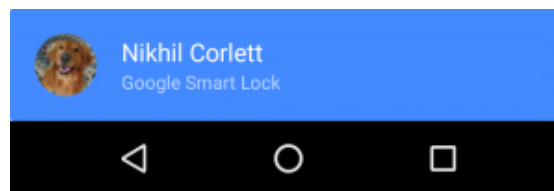
Smart Lock can also help speed up the process of signing up new users, allowing us to prompt the user to choose an existing account on the device and use any additional information from this account to pre-populate sign up forms. If a Google account is chosen, this can even be used to provide a federated login as detailed earlier.

According to Google, since implementing Smart Lock in their app, The New York Times has seen 80% of new logins assisted by Smart Lock. Netflix have also seen big benefits from Smart Lock, with a 20% reduction in support cases related to account recovery for Android users. Google claim that over 60% of users are opting to save their login credentials in major Smart Lock enabled apps, quite an impressive feat for a relatively young service.

The benefits of Smart Lock are quite clear, but what about the effort involved? Google does a lot of the heavy lifting for us by way of the Google Play Services library. This includes showing popups for account selection, saving credentials and sign in confirmation, along with synchronisation of

On iOS, despite less uptake of iCloud Keychain the feature shouldn't take much more effort to implement than Smart Lock on Android, and is still worth considering. We typically use the standard Keychain to store user credentials offline. From here, it is not too complex to synchronise these credentials, however there is no standard UI for selecting accounts. A custom selection view would have to be written as part of the login screen. Despite the extra effort, the improved user experience during the signup process will hopefully make it a worthwhile investment.

For more on Smart Lock, check out this post on the Android Developers blog.

The Distance can help implement social sign on and Smart Lock into your app, get in touch to find out how.

Janine Griffiths

We love to talk

Name *

Email *

Telephone

Message