

# Mobile client provisioning

The default sign-in flow in the mobile app requires the user to enter the correct account provider (homeserver) which is error prone and also not effortless.

In order to improve the user experience, and to skip this step, there are 3 options to specify the account provider or other configuration for the user:

- Use mobile device management (MDM).
- Use a deeplink to open the app.
- Use a custom app.

This is a new feature available in since 25.05.2 (iOS) and 25.06.0 (Android).

## Deeplinking

Deeplinking is the simplest option (unless the customer is already using MDM for their users or has a custom app).

## How does it work?

The customer would send an on-boarding email to its users which contains two major steps:

1. Download the app
2. Sign in to the app

The second step then contains the deeplink which opens the app and sets the desired account provider. Optionally, a QR code of the link would be included in case the user can't access the mail on their phone.

Here's an example of how such e-mail could look like. Note that currently ESS does not have means for composing and sending these emails to users automatically.

The configuration set by opening the app with the deeplink is retained only until the app is running. If the app is killed and opened without using the deeplink, the default behaviour will apply (e.g. user has to enter account provider).

If account provider is already set by MDM or customer publishes their own app which does not allow using an arbitrary account provider, the account provider in the deeplink is ignored.

If the user tries to open the app with the deeplink, but does not have the app installed yet on their phone, the user lands on a web page that guides them to install it.

# How to Create the Deeplink?

The format for the deeplink is:

```
https://mobile.element.io/<app>/?account_provider=<example.com>&login_hint=<alice>
```

The `<app>` specifies the mobile app to open:

- `element` -> Element X
- `element-pro` -> Element Pro
- `missing` -> Element classic

Note: while Element X is supported, it is expected that customers use the Element Pro app.

The `<example.com>` specifies the server name of the account provider aka homeserver, e.g. `example.com`. Note that for backward compatibility purposes the `hs_url` works as an alternative for the Element (classic) app, but it expects an URL, e.g. `https://example.com`.

The `<alice>` is intended to populate the username for the login. Since this is implemented by providing it as the `login_hint` parameter in the OIDC protocol, the exact value/format of it depends on which means of authentication is used.

When MAS is used **without** an upstream identity provider (e.g. no SSO for users), the format `mxid:@alice:example.org` is expected. An example deeplink/URL for `@alice:example.com` to sign to Element Pro in such a case is: `https://mobile.element.io/element-pro/?account_provider=example.com&login_hint=mxid:@alice:example.org`

When MAS is used **with** an upstream identity provider, you need to set `forward_login_hint: true` in your MAS configuration for the hint to be forwarded. See the [MAS documentation](#) for more details. The hint itself depends on what the provider expects, but in many cases the email address is used as the user ID, e.g. `alice@example.com`. Also, in case of SSO providing the username is less critical, as user is very likely already signed in because they have used other apps with the SSO.

To create a QR code of the link, online services can be used.

## App Download Links

The links to download the app are:

- Element Pro
  - iOS: <https://apps.apple.com/app/element-pro-for-work/id6502951615>
  - Android: <https://play.google.com/store/apps/details?id=io.element.enterprise>
- Element X

- iOS: <https://apps.apple.com/ee/app/element-x-secure-chat-call/id1631335820>
- Android: <https://play.google.com/store/apps/details?id=io.element.android.x>

Please refer to Apple and Google documentation how to use badges for the download buttons.

## Note: Redirect from Element Web

Note: when a user tries to access Element web app with a browser on a mobile device (e.g. they go to <https://chat.example.com>), the user is directed to a page which also guides them to download the native mobile app and has a button to open the app with the deeplink. In such a case the account provider is set to the one that the Element web installation specifies.

# Mobile Device Management (MDM)

If the customer is using MDM, the account provider can be delivered to the user's phone via the AppConfig.

When applying the MDM configuration to Element Pro, the Application/Bundle ID is `io.element.enterprise`. The key for specifying the account provider is `accountProvider`. The value must be the server name, e.g. `example.com`.

The end user just needs to download the app (if not installed by the MDM solution already) and open it.

Note that when MDM is used for specifying the account provider, other means like deeplink are ignored.

## Custom app

If the customer is publishing their own custom app in the stores, the account provider(s) is built into the app.

The account provider(s) are specified by the `accountProviders` parameter of the custom app's build pipeline (by Element who sets up the pipeline). Details are provided in the pipeline inputs documentation.

With custom app it is possible to specify more than one account provider. In such a case the user would still need to make a selection but would not need to input manually.

Note that the account provider(s) in the custom app are overridden by the one from MDM (if exists).

If a custom app is opened with a deeplink that also specifies account provider, the one in the deeplink is only applied when the `accountProviders` parameter of the custom app contains `*`. The latter means that users of the custom app are allowed to use arbitrary account providers.

---

