This article shows you how to authenticate users in a multi-tenant Identity Platform environment.

Make sure you've enabled multi-tenancy for your project and configured your tenants. See Getting started with multi-tenancy (/identity-platform/docs/multi-tenancy-quickstart) to learn how.

You'll also need to add the Client SDK to your application:

1. Go to the Identity Platform page in the Cloud Console.
   Go to the Identity Platform users page (https://console.cloud.google.com/customer-identity/users)
2. On the top right, click **Application setup details**.
3. Copy the code into your web app. For example:

To sign in to a tenant, the tenant ID needs to be passed to the `auth` object. Note that `tenantId` is not persisted on page reloads.

Any future sign-in requests from this `auth` instance will include the tenant ID (`TENANT_ID1` in the example above) until you change or reset the tenant ID.

You can work with multiple tenants using single or multiple `auth` instances.

To use a single `auth` instance, modify the `tenantId` property whenever you want to switch between tenants. To revert to back to project-level IdPs, set `tenantId` to `null`:

To use multiple instances, create a new `auth` instance for each tenant and assign them different IDs:

After signing in with a tenant, a tenant user will be returned with `user.tenantId` set to that tenant. Note that if you switch `tenantId` on the `auth` instance later, the `currentUser` property will not change; it will still point to the same user as the previous tenant.

The following example shows how to register a new user:

To sign in an existing user:

To sign in with a SAML provider, instantiate a `SAMLAuthProvider` instance with the provider ID from the Cloud Console:

You can then use either a popup or a redirect flow to sign in to the SAML provider.

In both cases, be sure to set the correct tenant ID on the `auth` instance.

To initiate the authentication flow, display an interface prompting the user to provide their email address, then call `sendSignInLinkToEmail` to send them an authentication link. Make sure to set the correct tenant ID on the `auth` instance before sending the email.

To complete sign-in on the landing page, first parse the tenant ID from the email link and set it on the `auth` instance. Then call `signInWithEmailLink` with the user's email and the actual email link containing the one-time code.

Creating a multi-tenant aware custom token is identical to creating a regular custom token; as long as the correct tenant ID has been set on the `auth` instance, a top-level `tenant_id` claim will be added to the resulting JWT. See Creating custom tokens (https://firebase.google.com/docs/auth/admin/create-custom-tokens) for detailed instructions on how to create and use custom tokens.

The following example shows how to create a custom token using the Admin SDK:

And the code below demonstrates how to sign in using a custom token:

Note that if the tenant IDs do not match, the `signInWithCustomToken()` method will fail.

You can link other types of credentials to an existing multi-tenant user. For example, if a user previously authenticated with a SAML provider in a tenant, you can add email/password sign-in to their existing account so they can use either method to sign in to the tenant.

When linking or re-authenticating an existing multi-tenant user, `auth.tenantId` will be ignored; use `user.tenantId` to specify which tenant to use. This also applies to other user management APIs, such as `updateProfile` and `updatePassword`.

If you enabled the **Link accounts that use the same email** setting in Cloud Console, when a user tries to sign in to a provider (such as SAML) with an email that already exists for another provider (such as Google), the error `auth/account-exists-with-different-credential` is thrown (along with an `AuthCredential` object).

To finish signing in with the intended provider, the user must first sign in to the existing provider (Google), then link to the former `AuthCredential` (SAML).

You can use either a popup or redirect flow to handle this error.

- Create a sign-in page for multiple tenants (/identity-platform/docs/multi-tenancy-ui)
- Migrate existing users to a tenant (/identity-platform/docs/migrate-users-between-projects-tenants)
- Manage tenants programmatically (/identity-platform/docs/multi-tenancy-managing-tenants)