Identity Platform allows users to authenticate to your applications and services, like multi-tenant SaaS applications, mobile/web apps, games, APIs and more. Identity Platform provides secure, easy-to-use authentication if you're building a service on Google Cloud, on your own backend or on another platform.

Identity Platform provides backend services and works with the easy-to-use SDKs and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook, Twitter, and any provider that supports SAML or OpenID Connect protocol.

Identity Platform integrates tightly with Google Cloud services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

You can use the SDK to integrate one or more sign-in methods into your app.

Authentication using the SDK

Email and password based authentication	Authenticate users with their email addresses and passwords. The SDK pro methods to create and manage users that use their email addresses and p to sign in. Identity Platform also handles sending password reset emails. iOS (https://firebase.google.com/docs/auth/ios/password-auth) Android (https://firebase.google.com/docs/auth/android/password-auth Web (https://firebase.google.com/docs/auth/web/password-auth) C++ (https://firebase.google.com/docs/auth/cpp/password-auth) Unity (https://firebase.google.com/docs/auth/unity/password-auth)
Federated identity provider integration	Authenticate users by integrating with federated identity providers. The SD methods that allow users to sign in with their Google, Facebook, Twitter, ar accounts.

	Google	iOS (https://firebase.google.com/docs/auth/ios/google-signin)
		Android (https://firebase.google.com/docs/auth/android/goo
		Web (https://firebase.google.com/docs/auth/web/google-sigr
		C++ (https://firebase.google.com/docs/auth/cpp/google-signi
		Unity (https://firebase.google.com/docs/auth/unity/google-si
		iOS (https://firebase.google.com/docs/auth/ios/facebook-login
		Android (https://firebase.google.com/docs/auth/android/face
	Facebook	Web (https://firebase.google.com/docs/auth/web/facebook-lc
		C++ (https://firebase.google.com/docs/auth/cpp/facebook-log
		Unity (https://firebase.google.com/docs/auth/unity/facebook-
		iOS (https://firebase.google.com/docs/auth/ios/twitter-login)
		Android (https://firebase.google.com/docs/auth/android/twit
	Twitter	Web (https://firebase.google.com/docs/auth/web/twitter-logi
		C++ (https://firebase.google.com/docs/auth/cpp/twitter-login
		Unity (https://firebase.google.com/docs/auth/unity/twitter-lo
		iOS (https://firebase.google.com/docs/auth/ios/github-auth)
		Android (https://firebase.google.com/docs/auth/android/gith
	GitHub	Web (https://firebase.google.com/docs/auth/web/github-auth
		C++ (https://firebase.google.com/docs/auth/cpp/github-auth)
		Unity (https://firebase.google.com/docs/auth/unity/github-au
Phone number authentication	Authenticate users by sending SMS messages to their phones.	

iOS (https://firebase.google.com/docs/auth/ios/phone-auth)

Android (https://firebase.google.com/docs/auth/android/phone-auth)

Web (https://firebase.google.com/docs/auth/web/phone-auth)

	C++ (https://firebase.google.com/docs/auth/cpp/phone-auth)
	Unity (https://firebase.google.com/docs/auth/unity/phone-auth)
Custom auth system integration	Connect your app's existing sign-in system to Identity Platform, exchanging generated on your server for Identity Platform tokens that can be used for running in Google Cloud, Firebase, or other services. iOS (https://firebase.google.com/docs/auth/ios/custom-auth) Android (https://firebase.google.com/docs/auth/android/custom-auth) Web (https://firebase.google.com/docs/auth/web/custom-auth) C++ (https://firebase.google.com/docs/auth/cpp/custom-auth) Unity (https://firebase.google.com/docs/auth/unity/custom-auth)
Anonymous auth	Use features that require authentication without requiring users to sign in f creating temporary anonymous accounts. If the user later chooses to sign upgrade the anonymous account to a regular account, so the user can con they left off. iOS (https://firebase.google.com/docs/auth/ios/anonymous-auth) Android (https://firebase.google.com/docs/auth/android/anonymous-au Web (https://firebase.google.com/docs/auth/web/anonymous-auth) C++ (https://firebase.google.com/docs/auth/cpp/anonymous-auth) Unity (https://firebase.google.com/docs/auth/unity/anonymous-auth)

To sign a user into your app, you first get authentication credentials from the user. These credentials can be the user's email address and password, a SAML assertion, or an OAuth token from a federated identity provider. In the case of federated identity providers, the providers return those tokens to Identity Platform's authentication handler on the `/__auth/handler` endpoint. This service is hosted by Google, so you don't have to receive and validate the

authentication artifact. After the tokens are received, our backend services will verify them and return a response to the client.

After a successful sign in, you can access the user's basic profile information, and you can control the user's access to data stored in Google Cloud or other products. You can also use the provided authentication token to verify the identity of users in your own backend services.

Using the SDK

Set up sign-in methods	For email address and password or phone number sign-in and any federated identity providers you want to support, enable them in the Cloud Console and complete any configuration required by the identity provider, such as setting your OAuth redirect URL.
Implement UI flows for your sign-in methods	For email address and password sign-in, implement a flow that prompts users to type their email addresses and passwords. For phone number sign-in, create a flow that prompts users for their phone number, and then for the code from the SMS message they receive. For federated sign-in, implement the flow required by each provider.
Pass the user's credentials to the SDK	Pass the user's email address and password or the OAuth token that was acquired from the federated identity provider to the SDK.

Using open source UI componenets

Set up sign-in methods	For email address and password or phone number sign-in and any federated identity providers you want to support, enable them in the or Cloud Console and complete any configuration required by the identity provider, such as setting your OAuth redirect URL.
Customize the sign-in UI	You can customize the sign-in UI by setting FirebaseUI options. To customize even further, fork the code on GitHub.
Use FirebaseUI to perform the sign-in flow	Import the FirebaseUI library, specify the sign-in methods you want to support, and initiate the FirebaseUI sign-in flow.

Learn more about users in a Identity Platform project, then see the integration guides for the sign-in providers you want to support:

OS (https://firebase.google.com/docs/auth/ios/start)

Android (https://firebase.google.com/docs/auth/android/start)

Web (https://firebase.google.com/docs/auth/web/start)

C++ (https://firebase.google.com/docs/auth/cpp/start)

Unity (https://firebase.google.com/docs/auth/unity/start)

Admin (https://firebase.google.com/docs/auth/admin)

OIDC (Web) (/identity-platform/docs/how-to-enable-application-for-oidc)

SAML (Web) (/identity-platform/docs/how-to-enable-application-for-saml)