

# Web3Auth Core Kit Single Factor Auth Web SDK

Web3Auth's `@web3auth/single-factor-auth` SDK is simple and easy-to-use SDK, which helps you implement the Web3Auth without redirection in a single key pair flow.

With the Web3Auth Plug and Play Web SDKs you can easily and quickly authenticate users and reconstruct their private key. Web3Auth Plug and Play Web SDK redirects users to a Web3Auth-hosted screen (i.e. <https://auth.web3auth.io>). This flow is great for most use cases. Still, sometimes you may want to customize the authentication flow such that you can have more control over the UI and UX of the authentication process (i.e. to avoid the redirection). This can be achieved through our new `@web3auth/single-factor-auth` SDK, where now, you'll be able to use Web3Auth and avoid the redirection to OpenLogin-hosted screens.

This SDK gives you semi-custodial keys. The `@web3auth/single-factor-auth` SDK gives you the simplest flow for onboarding a user with Web3Auth. The user just needs to click on the login button, connect to their favorite social account (OAuth) or any passwordless flow, like email-passwordless, biometric authentication etc., and a private key is generated for them. This flow has a 1/1 Key Share setup and the whole key via Web3Auth Auth Network key shares is provided directly to your app, and reconstructed within it.

This flow is still noncustodial since the nodes in the Web3Auth network have a 5/9 consensus mechanism to generate the key shares. However, one can claim that this flow is custodial to the OAuth Login provider since this is directly generated using credentials provided by them.

Hence we can call this flow a semi-custodial flow, and encourage users to enable MFA to have a 2/3 Key Share setup, which is fully non-custodial.

**This Documentation is based on the 7.0.1**

SDK Version. [â](#)

## Requirements [â](#)

- This is a frontend SDK and can only run in a browser environment
- Basic knowledge of JavaScript
- Supports all major JavaScript Frameworks, Libraries and Bundlers

note The minimum [pricing plan](#) to use this SDK in a production environment is the Growth Plan since custom verifier setup is a needed feature for Core Kit SDKs. You can use this SDK with all features enabled in the development environment for free.

## Resources [â](#)

- [Quick Start](#)
- : Get Started with an easy-to-follow integration of Web3Auth
- [Example Applications](#)
- : Explore our example applications and try the SDK yourself.
- [Troubleshooting](#)
- : Find quick solutions to common issues faced by developers.
- [Source Code](#)
- : Web3Auth is open sourced. You can find the source code on our GitHub repository.
- [Community Support Portal](#)
- : Join our community to get support from our team and other developers.

One Key Flow Guide For an optimal flow that allows users to create an MFA setup if desired, use a combination of Web3Auth Single Factor Auth Web SDK and Web3Auth No Modal SDK for a one-key flow without redirection.

## Helper SDKs [â](#)

### Provider packages [â](#)

For making RPC calls within your dApp, Web3Auth exposes respective providers for different chains. This provider can be used to interact with the connected chain using exposed functions within the provider. Currently, Web3Auth supports providers for both EVM and Solana chains. For other chains, one can easily get the private key from the Web3Auth SDK.

tip Check out the [Providers SDK Reference](#) to learn more.

- [For EVM based Chains @web3auth/ethereum-provider](#)

- [For Solana Blockchain@web3auth/solana-provider](#)
- [Common Key Provider for other Blockchains@web3auth/base-provider](#)

## Common Types and Interfaces<sup>â</sup>

[@web3auth/base](#)

<sup>â</sup>

This package gives access to common types and interfaces for Web3Auth. This comes in handy by providing you with a standard way of importing the values you need to work with the SDKs. We highly recommend using it while working with Typescript. [Edit this page](#) [Previous Multi Factor Authentication](#) [Next Install](#)