D4D4D4;--ch-t-background: #1E1E1E;--ch-t-lighter-inlineBackground: #1e1e1ee6;--ch-t-editor-background: #1E1E1E;--ch-t-editor-foreground: #D4D4D4;--ch-t-editor-rangeHighlightBackground: #ffffff0b;--ch-t-editor-infoForeground: #3794FF;--ch-t-editor-selectionBackground: #264F78;--ch-t-focusBorder: #007FD4;--ch-t-tab-activeBackground: #1E1E1E;--ch-t-tab-activeForeground: #ffffff;--ch-t-tab-inactiveBackground: #2D2D2D;--ch-t-tab-inactiveForeground: #ffffff80;--ch-t-tab-border: #252526;--ch-t-tab-activeBorder: #1E1E1E;--ch-t-editorGroup-border: #444444;--ch-t-editorGroupHeader-tabsBackground: #252526;--ch-t-editorLineNumber-foreground: #858585;--ch-t-input-background: #3C3C3C;--ch-t-input-foreground: #D4D4D4;--ch-t-icon-foreground: #C5C5C5;--ch-t-sideBar-background: #252526;--ch-t-sideBar-foreground: #D4D4D4;--ch-t-sideBar-border: #252526;--ch-t-list-activeSelectionBackground: #094771;--ch-t-list-activeSelectionForeground: #fffffe;--ch-t-list-hoverBackground: #2A2D2E; }

Privy Signer

In this guide, you will learn how to create a Privy(opens in a new tab) signer that can be added as a Safe owner and used to initialize any of the kits from the Safe{Core} SDK.

Check out the Safe Signers demo app(opens in a new tab) on GitHub to follow along this guide with the completed code.

Prerequisites

- Node.js and npm(opens in a new tab)
- .
- A Privy account(opens in a new tab)
- and an App ID.

Install dependencies

npm yarn pnpm _10 npm install @privy-io/react-auth

Steps

Imports

Here are the necessary imports for this guide.

_10 import { PrivyProvider, usePrivy, useWallets } from '@privy-io/react-auth' In addition, you will need to import a web3 library of your choice to use in the "Get the provider and signer" section. In this guide, we are usingviem .

Get the App ID

Check Privy documentation on how to create a new project in theidashboard(opens in a new tab) andget the client ID(opens in a new tab) .

Once you have it, you need to initialize thePRIVY_APP_ID variable with it.

_10 const PRIVY_APP_ID = // ...

Initialize Privy

Privy works with React hooks. This means you can wrap your app with thePrivyProvider and have access to several react hooks likeusePrivy() anduseWallets() that will provide all the functionality.

ProvyProvider receives anappId and aconfig object with different properties. CheckUsing the Privy React SDK(opens in a new tab) from Privy documentation to learn more about all the different configuration options, appearance, login methods, etc.

_10 <PrivyProvider _10 appId={PRIVY_APP_ID} _10 config={{ _10 embeddedWallets: { _10 createOnLogin: 'users-without-wallets' // defaults to 'off' _10 } _10 }} _10

> _10 _10 In this guide you will use the following variables and methods from theusePrivy() anduseWallets() React hooks.

_10 const { login, logout, ready, authenticated } = usePrivy() _10 const { ready: readyWallets, wallets } = useWallets()

Login

To login with an email address or social account you need to call the following method, that will open the popup and request the user to submit the login form.

_10 login()

Get the provider and signer

Once the user is logged in, you can get theprovider andsigner , which is the externally-owned account of the user that was derived from its credentials.

To do that there is auseEffect() that is executed when any of theready ,authenticated ,readyWallets andwallets variables have its value updated. Once they all aretrue andwallets has a length greater than zero, you have access to thewallets first element, which is the user's connected signer.

viem You can instantiate the provider usingviem and the following imports:

_10 import { createWalletClient, custom } from 'viem' _10 import { sepolia } from 'viem/chains' _15 useEffect(() => { _15 const init = async () => { _15 if (ready && authenticated && readyWallets && wallets.length > 0) { _15 const ethereumProvider = await wallets[0].getEthereumProvider() _15 _15 const provider = createWalletClient({ _15 chain: sepolia, _15 transport: custom(ethereumProvider) _15 }) _15 _15 const signer = wallets[0].address _15 } _15 } _15 init() _15 }, [ready, authenticated, readyWallets, wallets]) With theprovider andsigner you are ready to instantiate any of the kits from the Safe{Core} SDK and set up or use this signer as a Safe owner.

Logout

Finally, to logout the user, call thelogout() method.

_10 logout()

Recap and further reading

After following this guide, you are able to create a Safe signer using Privy and get theprovider andsigner required to initialize the kits from the Safe{Core} SDK.

Learn more about Privy by checking the following resources:

- Privy website(opens in a new tab)
- Privy documentation(opens in a new tab)
- Privy quickstart guide(opens in a new tab)