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; }

Dynamic Signer

In this guide, you will learn how to create a Dynamic(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.

Note that this guide will focus on supporting Ethereum and EVM-compatible wallets. You can enable, import and add more connectors(opens in a new tab) as needed.

Prerequisites

- A Dynamic account(opens in a new tab)
- and an environment ID.

Install dependencies

npm yarn pnpm _10 npm install @dynamic-labs/sdk-react-core @dynamic-labs/ethereum

Steps

Imports

Here are the necessary imports for this guide.

```
_10 import { _10 DynamicContextProvider, _10 useDynamicContext, _10 useIsLoggedIn _10 } from '@dynamic-labs/sdk-react-core' _10 import { _10 EthereumWalletConnectors, _10 isEthereumWallet _10 } from '@dynamic-labs/ethereum'
```

Get the Environment ID

Once you create a new Dynamic project in the dashboard(opens in a new tab), you will find your environment ID in the SDK & API Keys page(opens in a new tab) .

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

```
_10 const DYNAMIC_ENVIRONMENT_ID = // ...
```

Initialize Dynamic

Dynamic works with React hooks. This means you can wrap your app with theDynamicContextProvider and have access to several react hooks likeuseDynamicContext() that will provide all the functionality.

DynamicContextProvider receives anenvironmentId and other configuration options. Check Dynamic React SDK documentation(opens in a new tab) to learn more about all the different configuration options.

```
_10 <DynamicContextProvider _10 settings={{ _10 environmentId: DYNAMIC_ENVIRONMENT_ID, _10 walletConnectors: [EthereumWalletConnectors], _10 // Add other configuration options as needed _10 }} _10
```

> _10 _10 In this guide you will use theprimaryWallet from theuseDynamicContext() hook, and theuseIsLoggedIn() hook.

```
_10 const { primaryWallet } = useDynamicContext() _10 const isLoggedIn = useIsLoggedIn()
```

Login

In theApp component, add the DynamicWidget component(opens in a new tab) that will handle all the login and authentication logic for you.

```
_10 import { DynamicWidget } from '@dynamic-labs/sdk-react-core' _10 _10 const App = () => { _10 return ( _10 <> _10 // Your other content here _10 _10
```

```
_10 ) _10 }
```

Get the provider and signer

Once the user is logged in through theDynamicWidget , 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 executes when any of theisLoggedIn andprimaryWallet variables have their values updated. Once they are alltrue , you have access to the user's connected signer. Note that you should also check that the wallet is an Ethereum wallet using theisEthereumWallet() function.

You can instantiate the provider and signer like so:

```
_10 useEffect(() => { _10 const init = async () => { _10 if (isLoggedIn && primaryWallet && isEthereumWallet(primaryWallet)) { _10 const provider = await primaryWallet.getWalletClient() _10 _10 const signer = primaryWallet.address _10 } _10 } _10 init() _10 }, [isLoggedIn, primaryWallet])
```

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.

Recap and further reading

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

Learn more about Dynamic by checking the following resources:

- Dynamic website(opens in a new tab)
- Dynamic documentation(opens in a new tab)
- Dynamic quickstart guide(opens in a new tab)
- Dynamic in a Safe App(opens in a new tab)

Signers Magic Was this page helpful?

Report issue