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; }

Magic Signer

In this guide, you will learn how to create aMagic(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 theSafe 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)
- .
- AMagic account(opens in a new tab)
- and an API key.

Install dependencies

npm yarn pnpm _10 npm install magic-sdk

Steps

Imports

Here are the necessary imports for this guide.

_10 import { Magic } from 'magic-sdk' 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 API Key

Check Magic documentation on how to create a new project in theidashboard(opens in a new tab) andget the API key.(opens in a new tab)

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

```
_10 const MAGIC_API_KEY = // ...
```

Initialize Magic

Magic initialization is done by instantiating theMagic class with anMAGIC_API_KEY . Optionally, it also accepts some configuration parameters that can be checked in theMagic Web API Reference(opens in a new tab) .

```
_10 const magic = new Magic(MAGIC_API_KEY)
```

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 await magic.wallet.connectWithUI()
```

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.

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

_10 import { createWalletClient, custom } from 'viem' _10 import { sepolia } from 'viem/chains' _10 const provider = createWalletClient({ _10 chain: sepolia, _10 transport: custom(magic.rpcProvider) _10 }) _10 _10 const metadata = await magic.user.getInfo() _10 const signer = metadata.publicAddress 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 await magic.user.logout()
```

Recap and further reading

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

Learn more about Magic by checking the following resources:

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

Dynamic Passkeys Was this page helpful?

Report issue