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; }

Send Transactions

This guide will teach you how to deploy new Safe accounts and create, sign, and execute Safe transactions using the Safe React Hooks.

For more detailed information, see the Safe React Hooks Reference .

Prerequisites

- Node.js and npm(opens in a new tab)

Install dependencies

First, you need to install some dependencies.

```
_10 pnpm add @safe-global/safe-react-hooks
```

Steps

Imports

Here are all the necessary imports for this guide.

```
_10 import { _10 SafeProvider, _10 createConfig, _10 useSafe, _10 useSendTransaction, _10 SendTransactionVariables, _10 useConfirmTransaction, _10 ConfirmTransactionVariables _10 } from '@safe-global/safe-react-hooks' _10 import {
```

sepolia } from 'viem/chains'

Create a signer and provider

Firstly, you need to get a signer, which will be the owner of a Safe account after it's deployed.

This example uses a private key, but any way to get an EIP-1193 compatible signer can be used.

_10 const SIGNER_ADDRESS = // ... _10 const SIGNER_PRIVATE_KEY = // ... _10 _10 const RPC_URL = 'https://rpc.ankr.com/eth_sepolia'

Initialize the Safe React Hooks

You need to wrap your app with theSafeProvider to have access to the different Safe React Hooks likeuseSendTransaction() ,useConfirmTransaction() , andusePendingTransactions() that will provide the functionality you need in this guide.

SafeProvider receives aconfig object with different properties to create the global configuration that you can get from thecreateConfig function.

New Safe account Existing Safe account When deploying a new Safe account for the connected signer, you need to pass the configuration of the Safe in thesafeOptions property. In this case, the Safe account is configured with your signer as the only owner.

_10 const config = createConfig({ _10 chain: sepolia, _10 provider: RPC_URL, _10 signer: SIGNER_PRIVATE_KEY, _10 safeOptions: { _10 owners: [SIGNER_ADDRESS], _10 threshold: 1 _10 } _10 }) To apply the global configuration to your app, pass the createdconfig to theSafeProvider .

_10 _10 _10

Create a Safe transaction

Create an array of Safe transactions to execute.

_10 const transactions = [{ _10 to: '0x...', _10 data: '0x', _10 value: '0' _10 }]

Send the Safe transaction

Create aSendTransaction component in your application to create and send a transaction.

If you configured your Safe withthreshold equal to1 , calling thesendTransaction function from theuseSendTransaction hook will execute the Safe transaction. However, if thethreshold is greater than1 the other owners of the Safe will need to confirm the transaction until the required number of signatures are collected.

SendTransaction.tsx _15 function SendTransaction() { _15 const { sendTransaction } = useSendTransaction() _15 _15 const sendTransactionParams: SendTransactionVariables = { _15 transactions _15 } _15 _15 return (_15

sendTransaction(sendTransactionParams)}> _15 Send Transaction _15

_15) _15 } _15 _15 export default SendTransaction

Confirm the Safe transaction

Create aConfirmPendingTransactions component in your application to check the transactions pending for confirmation in case the Safe transaction needs to be confirmed by other Safe owners.

Retrieve all the pending Safe transactions from the Safe Transaction Service by calling thegetPendingTransaction function from theuseSafe hook, and call theconfirmTransaction function from theuseConfirmTransaction hook to confirm them.

Notice that theSafeProvider configuration needs to be initialized with a different Safe owner as thesigner when confirming a transaction.

ConfirmPendingTransactions.tsx _20 function ConfirmPendingTransactions() { _20 const { getPendingTransactions } = useSafe() _20 const { data = [] } = getPendingTransactions() _20 const { confirmTransaction } = useConfirmTransaction() _20 _20 return (_20 <> _20 {data.length > 0 && data.map(tx => (_20 <> _20 {tx.safeTxHash} _20 confirmTransaction({ _20 safeTxHash: tx.safeTxHash _20 })} /> _20

_20))} _20

_20) _20 } _20 _20 export default ConfirmPendingTransactions Once the total number of confirmations reaches thethreshold of the Safe, theconfirmTransaction will automatically execute the transaction.

Recap and further reading

After following this guide, you are able to deploy new Safe accounts and create, sign, and execute Safe transactions using the Safe React Hooks.

Safe React Hooks Reference Was this page helpful?

Report issue