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; }

Migrate to v5

This guide references the major changes between v4 and v5 to help those migrating an existing app.

RemovingSafeFactory

class

TheSafeFactory class, previously used for deploying Safes, has been removed. The functionality to deploy Safes is now directly available in theSafe class through the newcreateSafeDeploymentTransaction method.

Old Method UsingSafeFactory

_24 // old v4 code _24 import { SafeFactory, SafeAccountConfig } from '@safe-global/protocol-kit' _24 _24 const safeFactory = await SafeFactory.init({ _24 provider, _24 signer, _24 safeVersion // Optional _24 }) _24 _24 const safeAccountConfig: SafeAccountConfig = { _24 owners: ['0x...', '0x...', '0x...'], _24 threshold: 2 _24 } _24 _24 const protocolKit = await safeFactory.deploySafe({ _24 safeAccountConfig, _24 saltNonce // Optional _24 }) _24 _24 // Confirm the Safe is deployed and fetch properties _24 console.log('Is Safe deployed:', await protocolKit.isSafeDeployed()) _24 console.log('Safe Address:', await protocolKit.getAddress()) _24 console.log('Safe Owners:', await protocolKit.getOwners()) _24 console.log('Safe Threshold:', await protocolKit.getThreshold())

New Method UsingSafe

class

_45 // new v5 code _45 import Safe, { PredictedSafeProps } from '@safe-global/protocol-kit' _45 _45 const predictedSafe:

PredictedSafeProps = { _45 safeAccountConfig: { _45 owners: ['0x...', '0x...', '0x...'], _45 threshold: 2 _45 }, _45 safeDeploymentConfig: { _45 saltNonce, // Optional _45 safeVersion // Optional _45 } _45 } _45 _45 let protocolKit = await Safe.init({ _45 provider, _45 signer, _45 predictedSafe _45 }) _45 _45 // you can predict the address of your Safe if the Safe version is v1.3.0 or above _45 const safeAddress = await protocolKit.getAddress() _45 _45 const deploymentTransaction = await protocolKit.createSafeDeploymentTransaction() _45 _45 // Execute this transaction using the integrated signer or your preferred external Ethereum client _45 const client = await protocolKit.getSafeProvider().getExternalSigner() _45 _45 const txHash = await client.sendTransaction({ _45 to: deploymentTransaction.to, _45 value: BigInt(deploymentTransaction.value), _45 data: deploymentTransaction.data as 0x{string}, _45 chain: sepolia _45 }) _45 _45 const txReceipt = await client.waitForTransactionReceipt({ hash: txHash }) _45 _45 // Reconnect to the newly deployed Safe using the protocol-kit _45 protocolKit = await protocolKit.connect({ safeAddress }) _45 _45 // Confirm the Safe is deployed and fetch properties _45 console.log('Is Safe deployed:', await protocolKit.isSafeDeployed()) _45 console.log('Safe Address:', await protocolKit.getAddress()) _45 console.log('Safe Owners:', await protocolKit.getOwners()) _45 console.log('Safe Threshold:', await protocolKit.getThreshold())

Predict the Safe Address

You can predict the address of a Safe account before its deployment, as long as you are using Safev1.3.0 or greater, by replacing theSafeFactory.predictSafeAddress method with theSafe.getAddress method:

_10 // old v4 code _10 const predictedSafeAddress = await safeFactory.predictSafeAddress(_10 safeAccountConfig, _10 saltNonce // optional _10) _10 _10 // new v5 code _10 const predictedSafeAddress = await protocolKit.getAddress()

Migration Steps

- Remove any import or reference of theSafeFactory
- class from your code.
- Replace theSafeFactory.deploySafe
- method with theSafe.createSafeDeploymentTransaction
- method where necessary. You can use your Ethereum client to execute this deployment transaction.
- To predict the Address of your Safe Account, replace theSafeFactory.predictSafeAddress
- method with theSafe.getAddress
- method.
- After the deployment transaction has been executed, it is necessary to reconnect the Protocol Kit instance to the newly created Safe address by using theconnect
- method.

The removal ofSafeFactory means thereâs one less class to initialize and manage within your project. You can now directly use theSafe class to handle all operations related to Safes, including their deployment.

Migrate to v4 Reference Was this page helpful?

Report issue