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; }

Safe4337Pack

TheSafe4337Pack enables Safe accounts to interact with user operations through the implementation of theRelayKitBasePack . You can find more about ERC-4337 atthis link(opens in a new tab) .

Install dependencies

To useSafe4337Pack in your project, start by installing therelay-kit package with this command:

```
_10 yarn add @safe-global/relay-kit
```

Reference

TheSafe4337Pack class make easy to use theSafe 4337 Module(opens in a new tab) with your Safe. It enables creating, signing, and executing transactions grouped in user operations using a selected provider. You can select your preferredbundler(opens in a new tab) andpaymaster(opens in a new tab) .

```
_10 const safe4337Pack = await Safe4337Pack.init({ _10 ethersAdapter, _10 rpcUrl, _10 bundlerUrl, _10 safeModulesVersion, _10 customContracts, _10 options, _10 paymasterOptions _10 })
```

init(safe4337InitOptions)

The static methodinit() generates an instance ofSafe4337Pack . Use this method to create the initial instance instead of the regular constructor.

Parameters

TheSafe4337InitOptions used in theinit() method are:

_33 Safe4337InitOptions = { _33 ethersAdapter: EthersAdapter _33 bundlerUrl: string _33 rpcUrl: string _33 safeModulesVersion?: string _33 customContracts?: { _33 entryPointAddress?: string _33 safe4337ModuleAddress?: string _33 addModulesLibAddress?: string _33 } _33 options: ExistingSafeOptions | PredictedSafeOptions _33 paymasterOptions?: PaymasterOptions _33 } _33 _33 ExistingSafeOptions = { _33 safeAddress: string _33 } _33 _33 PredictedSafeOptions = { _33 owners: string[] _33 threshold: number _33 safeVersion?: SafeVersion _33 saltNonce?: string _33 } _33 _33 PaymasterOptions = { _33 paymasterUrl?: string _33 isSponsored?: boolean _33 sponsorshipPolicyId?: string _33 paymasterAddress: string _33 paymasterTokenAddress?: string _33 amountToApprove?: bigint _33 }

- ethersAdapter
- : An instance of theEthersAdapter
- class.
- rpcUrl
- : The RPC for the selected chain.
- bundlerUrl
- : The bundler's URL.
- safeModulesVersion
- : The version of theSafe Modules contract(opens in a new tab)
- .
- customContracts
- : An object with custom contract addresses. This is optional, if no custom contracts are provided, default ones will be used.* entryPointAddress
-
  - : The address of the entry point. Defaults to the address returned by theeth_supportedEntryPoints
-
  - method from the provider API.
-
  - safe4337ModuleAddress
-
  - : The address of theSafe4337Module
-
  - . Defaults tosafe-modules-deployments
-
  - using the current version.
-
  - addModulesLibAddress
-
  - : The address of theAddModulesLib
-
  - library. Defaults tosafe-modules-deployments
-
  - using the current version.
- options
- : The Safe account options.* safeAddress
-
  - : The Safe address. You can only use this prop to specify an existing Safe account.
-
  - owners
-
  - : The array with Safe owners.
-
  - threshold
-
  - : The Safe threshold. This is the number of owners required to sign and execute a transaction.
-
  - safeVersion
-
  - : The version of theSafe contract(opens in a new tab)
-
  - . Defaults to the current version.
-
  - saltNonce
-
  - : The Safe salt nonce. Changing this value enables the creation of different safe (predicted) addresses using the same configuration (owners
-
  - ,threshold

-
  - , andsafeVersion
-
  -).
- paymasterOptions
- : The paymaster options.* paymasterUrl
-
  - : The paymaster URL. You can obtain the URL from the management dashboard of the selected services provider. This URL will be used for gas estimations.
-
  - isSponsored
-
  - : A boolean flag to indicate if we want to use a paymaster to sponsor transactions.
-
  - sponsorshipPolicyId
-
  - : The sponsorship policy ID can be obtained from the management dashboard of the selected payment services provider.
-
  - paymasterAddress
-
  - : The address of the paymaster contract to use.
-
  - paymasterTokenAddress
-
  - : The paymaster token address for transaction fee payments.
-
  - amountToApprove
-
  - : ThepaymasterTokenAddress
-
  - amount to approve.

Returns A promise that resolves to an instance of theSafe4337Pack .

Caveats

- Use this method to create the initial instance instead of the standard constructor.
- You can refer tothis link(opens in a new tab)
- to create instances ofEthersAdapter
- .
- You should search for some API services URLs and contract addresses in the management dashboards of your selected provider. These includebundlerUrl
- ,paymasterUrl
- ,paymasterAddress
- ,paymasterTokenAddress
- ,sponsorshipPolicyId
- , andrpcUrl
- (In this case any valid RPC should be fine).
- The SDK uses default versions whensafeModulesVersion
- orsafeVersion
- are not specified. You can find more details about the current versionshere(opens in a new tab)
- .
- ThesaltNonce
- derives different Safe addresses by using theprotocol-kit
- methodpredictSafeAddress
- . You can find more details about this processhere(opens in a new tab)
- .
- We typically initialize the pack in two ways. One way is by using an existing account with thesafeAddress
- prop. The other way is by using theowners
- ,threshold
- ,saltNonce
- , andsafeVersion
- props to create a new Safe account. You can also apply the second method to existing addresses, as the output address will be the same if the inputs are identical.
- The SDK querieseth_supportedEntryPoints
- for a defaultentryPointAddress
- if not given. It fetchessafe4337ModuleAddress
- andaddModulesLibAddress

- from thesafe-modules-deployments
- repository if not provided. You can find them atsafe-modules-deployments(opens in a new tab)
- .
- To use a paymaster without sponsorship, you need to hold a certain amount ofpaymasterTokenAddress
- in the Safe account for fees. Make sure to provide thepaymasterAddress
- as well.
- You can choose to use a paymaster to sponsor transactions by setting theisSponsored
- prop. When sponsoring transactions, you need to provide thepaymasterUrl
- ,paymasterAddress
- , and optionally thesponsorshipPolicyId
- .
- An approval for the concrete ERC-20 token is required to use the paymaster so remember to add thepaymasterTokenAddress
- of the ERC-20 token that will pay the fees. The SDK will encode this approval internally and send it to the bundler with the rest of the user operation.
- Specify the amount to approve for thepaymasterTokenAddress
- using theamountToApprove
- prop. This is necessary when the Safe account is not deployed, and you need to approve the paymaster token for fee payments and Safe account setup.

new Safe4337Pack({protocolKit, bundlerClient, publicClient, bundlerUrl, paymasterOptions, entryPointAddress, safe4337ModuleAddress})

TheSafe4337Pack constructor method is used within theinit() method and should not be directly accessed. The parameters are calculated or provided by theinit() method.

createTransaction(safe4337CreateTransactionProps)

Create aSafeOperation from a transaction batch. You can send multiple transactions to this method. The SDK internally bundles these transactions into a batch sent to the bundler as aUserOperation . If the transaction is only one then no batch is created a it's not necessary.

Parameters

TheSafe4337CreateTransactionProps

_10 Safe4337CreateTransactionProps = { _10 transactions: MetaTransactionData[] _10 options?: { _10 amountToApprove?: bigint _10 validUntil?: number _10 validAfter?: number _10 feeEstimator?: IFeeEstimator _10 } _10 }

- transactions
- : Array ofMetaTransactionData
- to batch in aSafeOperation
- (using the multisend contract if more than one transaction is included).
- options
- : Optional parameters.* amountToApprove
-
    - : The amount to approve to thepaymasterTokenAddress
-
    - .
-
    - validUntil
-
    - : The UserOperation will remain valid until this block's timestamp.
-
    - validAfter
-
    - : The UserOperation will be valid after this block's timestamp.
-
    - feeEstimator
-
    - : The fee estimator calculates gas requirements by implementing theIFeeEstimator
-
    - interface.

Returns A promise that resolves to theSafeOperation .

Caveats

- TheSafeOperation

- is similar to the standard user operation but includes Safe-specific fields. Before sending it to the bundler, we convert theSafeOperation
- to a regular user operation. We need to sign the operation for the bundler to execute it using theSafe4337Module
- .
- You can set theamountToApprove
- in this method to approve thepaymasterTokenAddress
- for transaction payments, similar to howamountToApprove
- works in theinit()
- method.
- We use a similar API toprotocol-kit
- for developers transitioning toSafe4337Pack
- . This API helps with creating and executing transactions, bundling user operations and sending them to the bundler.
- UsevalidUntil
- andvalidAfter
- to set the block timestamp range for the user operation's validity. The operation will be rejected if the block timestamp falls outside this range.
- ThefeeEstimator
- calculates gas needs for the UserOperation. We default to Pimlico'sfeeEstimator
- , but you can use a different one by providing your own. The IFeeEstimator interface requires an object with specific methods.

_18 IFeeEstimator { _18 setupEstimation?: EstimateFeeFunction _18 adjustEstimation?: EstimateFeeFunction _18 getPaymasterEstimation?: EstimateSponsoredFeeFunction _18 } _18 _18 EstimateFeeFunctionProps = { _18 userOperation: UserOperation _18 bundlerUrl: string _18 entryPoint: string _18 } _18 _18 EstimateSponsoredFeeFunctionProps = { _18 userOperation: UserOperation _18 paymasterUrl: string _18 entryPoint: string _18 sponsorshipPolicyId?: string _18 }

All methods are optional and will be called in the specified order if you provide any of them:

1. setupEstimation
2. : This method, called before using the bundlereth_estimateUserOperationGas
3. in the pack code, allows you to adjust the user operation before the bundler estimates it, as each provider has its own recommendations.
4. adjustEstimation
5. : This method is used after callingeth_estimateUserOperationGas
6. in the pack code to adjust the bundler estimation.
7. getPaymasterEstimation
8. : After using the bundlereth_estimateUserOperationGas
9. from the package code, this method is used if the user operation is sponsored. It helps adjust the bundler's estimation when a paymaster sponsors the transaction that use to involve some specific fee estimations.

signSafeOperation(safeOperation, signingMethod)

Signs aSafeOperation .

Parameters

- safeOperation
- : TheSafeOperation
- to sign.
- signingMethod
- : The method to use for signing the transaction. The default isSigningMethod.ETH_SIGN_TYPED_DATA_V4
- .

Returns A promise that resolves to the signedSafeOperation .

Caveats

- Use this method after theSafeOperation
- is generated with thecreateTransaction
- method.
- This method adds the signer's signature from theEthersAdapter
- to the signatures map of theSafeOperation
- object. Additional signatures can be included from multiple owners.
- It works similar tosignTransaction
- andsignMessage
- methods in theprotocol-kit
- but usingSafeOperation
- instead ofSafeTransaction
- orSafeMessage

- . For more information, refer to the Safedocs(opens in a new tab)
- .

executeTransaction(safe4337ExecutableProps)

This method sends the user operation to the bundler.

â¹ï¸ If you are not using a paymaster and need to deploy a new Safe (counterfactual deployment), you must hold in the predicted Safe address the amount of native token required to cover the fees.

Parameters

TheSafe4337ExecutableProps

_10 Safe4337ExecutableProps = { _10 executable: SafeOperation _10 }

- executable
- : TheSafeOperation
- to execute.

Returns A promise, resolves to the user operation hash.

Caveats

- The process converts theSafeOperation
- to a standard user operation, then forwards it to the bundler. TheSafeOperation
- must be created and signed by the Safe owner usingEthersAdapter
- .
- You can use the user operation hash to browse the status (e.ghttps://jiffyscan.xyz/userOpHash/{userOpHash}
-)

getUserOperationByHash(userOpHash)

Retrieve the user operation using its hash.

Parameters

- userOpHash
- : The user operation hash is returned by theexecuteTransaction
- method. The user operation can be executed or pending, and the method will return the payload data for the user operation.

Returns A Promise that resolves toUserOperationWithPayload .

_10 UserOperationWithPayload = { _10 userOperation: UserOperation _10 entryPoint: string _10 transactionHash: string _10 blockHash: string _10 blockNumber: string _10 }

Caveats

- Use this method to request information about the user operation sent to the bundler, but do not use it for the execution status.

getUserOperationReceipt(userOpHash)

GetUserOperation receipt by a hash.

Parameters

- userOpHash
- : Unique identifier for theUserOperation

Returns A Promise that resolves toUserOperationReceipt after the user operation is executed.

_10 UserOperationReceipt = { _10 userOpHash: string _10 sender: string _10 nonce: string _10 actualGasUsed: string _10 actualGasCost: string _10 success: boolean _10 logs: Log[] _10 receipt: Receipt _10 }

Caveats

- Use this method to obtain the full execution trace and status.
- You can use this method to check if theUserOperation
- was successful by calling it repeatedly until the receipt is available.

getSupportedEntryPoints()

Retrieve all supported entry points.

Returns A promise that resolves to an array of entry point addresses (strings) supported by the bundler.

Caveats We use this method to obtain the default entry point if not provided in theinit() method.

getChainId()

Retrieve the EIP-155 Chain ID.

Returns A promise that resolves to the EIP-155 Chain ID string.

Gelato Relay Migrating to V2