

# SafeAuthPack

SafeAuth is a pluggable authentication infrastructure for web3 wallets and applications. It simplifies onboarding for mainstream and crypto native users, offering experiences tailored to their preferences. It supports all social logins, web and mobile native platforms, wallets, and other key management methods, creating a cryptographic key provider specific to the user and application.

TheSafeAuthPack enables using the [Web3Auth infrastructure\(opens in a new tab\)](#) with added Safe capabilities, such as retrieving the related Safe addresses for a given externally-owned account created using SafeAuth services.

The pack helps onboard web2 users and provides an Ethereum signer address that can be used across any dapp using the different Ethereum chains.

## Install dependencies

To use theSafeAuthPack , you must install the@safe-global/auth-kit package and the corresponding Web3Auth one.

```
yarn
add
@safe-global/auth-kit
@web3auth/safeauth-embed
```

## Reference

TheSafeAuthPack class makes SafeAuth modal and Safe accounts work together. Create an instance of the pack and initialize it to start the interaction.

```
const
safeAuthPack
=
new
SafeAuthPack ({ txServiceUrl :
'https://safe-transaction-mainnet.safe.global' , }); await
safeAuthPack .init (safeAuthInitOptions);
```

### new SafeAuthPack(safeAuthConfig)

Parameters

- safeAuthConfig?
- - TheSafeAuthPack
- class instantiation accepts the following options in its configuration:

SafeAuthConfig { txServiceUrl ? : string }

- txServiceUrl?
- - The URL of the Safe Transaction Service. It's mandatory in chains where Safe doesn't provide a Transaction Service. It retrieves the Safe addresses for an EOA created using SafeAuth services.

Caveats You should always call theinit() method afterward before interacting with the pack.

### init(safeAuthInitOptions)

The init method initializes the provided Web3Auth SDK and Safe services. It creates an embedded browser wallet within an iframe, establishing communication through the internally generated EIP-1193 provider.

Parameters

- safeAuthInitOptions
- - The options to initialize the SDK instance.

safeAuthInitOptions { enableLogging ? : boolean showWidgetButton ? : boolean buttonPosition ? :

"bottom-left"

|

"top-left"

|

"bottom-right"

|

"top-right" buildEnv ? :

"production"

|

"development"

|

"testing" chainConfig ? : { blockExplorerUrl : string logo : string tickerName : string ticker : string rpcTarget : string wcTarget ? : string chainId : string displayName : string isTestnet ? : boolean isErc20 ? : boolean tokenAddress ? : string } }

- enableLogging
- - Enable logging for the SDK. Defaults to false
- .
- buildEnv
- - The build environment. production
- and testing
- use https://safe.web3auth.com
- . development
- use http://localhost:4050
- . Defaults to production
- .
- showWidgetButton
- - Show the widget button. Defaults to true
- .
- buttonPosition
- - If showWidgetButton
- is true then this prop represent the position of the widget button. Defaults to bottom-left
- .
- chainConfig
- - The chain configuration. Defaults to ethereum
- if no configuration is provided.\* blockExplorerUrl
- - - Block explorer URL of the chain (e.g. https://etherscan.io
- - ).
- - logo
- - - Logo URL of the base token of the chain (e.g. https://eth.svg
- - ).
- - tickerName

- - - Name for ticker (e.g Ethereum).
- - ticker
- - - Symbol for ticker (e.g ETH).
- - rpcTarget
- - - The RPC URL to be used.
- - wcTarget?
- - - The WebSocket URL to be used. Use this or rpcTarget
- - .
- - chainId
- - - The chain ID to be used. Should be an hex with 0x prefix (e.g 0x1 for Mainnet).
- - displayName
- - - The display name for the network.
- - isTestnet?
- - - Whether the network is Testnet or not.
- - isErc20?
- - - Whether the token is an ERC20 token or not.
- - tokenAddress?
- - - The token address for the chain. Should be an hex with 0x prefix (e.g 0x6b175474e89094c44da98b954eedeac495271d0f for DAI).

## Caveats

- The list of supported chains is:
  - \* Production: Ethereum, Polygon, BSC, Avalanche, Optimism, Celo, Arbitrum, Gnosis chain.
- - Test: Sepolia, Polygon Mumbai, BSC Testnet, Avalanche Testnet, Arbitrum Testnet, Optimism Testnet.
- Call always theinit()
- method before interacting with the other methods in the pack.
- Theinit()
- method creates an iframe and establishes a connection with the embedded wallet domain. To remove the iframe and disconnect the connection, use thesignOut()
- method. If you want to sign out and sign in again in a single-page application (SPA) fashion, avoid using signOut({ reset: true })
- as it will clean up the session, iframe, and connection. Instead, you will need to re-instantiate the pack.

## signIn(safeAuthSignInOptions?)

signIn(safeAuthSignInOptions) starts the authentication flow. It displays a popup that enables users to select a social authentication method (OAuth) or an email to generate the web3 wallet address. It returns the EOA and the associated Safe addresses.

## Parameters

# SafeAuthSignInOptions

{ loginProvider? :

"google"

|

"facebook"

|

"reddit"

|

"discord"

|

"twitch"

|

"apple"

|

"line"

|

"github"

|

"kakao"

|

"linkedin"

|

"twitter"

|

"weibo"

|

"wechat"

|

"email\_passwordless" login\_hint ? : string }

- loginProvider
- - If specified, instead of showing the popup to choose the OAuth or email, a direct attempt to login with the specified provider will be made.
- login\_hint
- - Used to provide default mail given a loginProvider
- :

safeAuthPack .signIn ({ loginProvider :

'google' , login\_hint :

'my-mail@safe.global' , });

Returns

An object with the derived externally-owned account address and the associated Safe addresses.

```
AuthKitSignInData { eoa : string safes ? : string[] }
```

Caveats

- To get the Safe addresses, instantiate the `authKit`
- with the `txServiceUrl`
- property in the config object. Otherwise, only the EOA will be returned.
- ⚠, This method currently returns the Safe addresses where the EOA is an owner. It doesn't create a Safe. We are investigating ways to enhance the Auth Kit and the associated flows.

## **signOut(safeAuthSignOutOptions?)**

Call this method to sign out the user and clean the session.

```
SafeAuthSignOutOptions { reset : boolean; }
```

Parameters

- `reset`
- - If true, the user will be logged out from the provider, and the widget will be destroyed. Don't use this parameter if you want to log out and log in again without refreshing the browser or re-initializing the `SafeAuthPack`
- `instance`.

## **getUserInfo()**

Using `getUserInfo()`, you will receive the user information derived from the pack you are using. It varies depending on the provider.

Returns

The `UserInfo` object's properties vary depending on the provider.

## **getProvider()**

By using `getProvider()`, you will receive a standard web3 provider compatible with the EIP-1193 standard.

Returns

A EIP-1193 compatible provider.

Caveats

- You can wrap the provider using your favorite library (ethers, web3, etc.).

## **destroy()**

This method removes the iframe. It's useful when you need to re-instantiate the pack, for example, when changing the connected chain.

## **subscribe(event, handler)**

Allow to subscribe to authentication state changes.

Parameters

- `event`
- - The event you want to subscribe to. Currently, you can subscribe to `accountsChanged`
- `chainChanged`
- `.`
- `handler`
- - When the event is triggered, the function will be called.

## **unsubscribe(event, handler)**

Allow to unsubscribe to authentication state changes.

### Caveats

The `accountsChanged` event helps detect whether the user has previously signed in. This allows you to reauthenticate when the browser is refreshed by calling the `signIn` method, preventing the user from clicking the sign-in button again.

### Parameters

- event
  - The event you want to unsubscribe to.
- handler
  - The function will be called when the event is triggered.

## Usage

Calling `init()` when your page loads or component renders is all it takes to use the Auth Kit with the `SafeAuthPack`. To start the authentication process, call `signIn()` afterward. This method returns the EOA and associated Safe addresses.

```
// Instantiate and initialize the pack const
```

```
safeAuthPack
```

```
=
```

```
new
```

```
SafeAuthPack (safeAuthConfig); await
```

```
safeAuthPack .init (safeAuthInitOptions);
```

```
const { eoa ,
```

```
safes } =
```

```
await
```

```
safeAuthPack .signIn (); const
```

```
userInfo
```

```
=
```

```
await
```

```
safeAuthPack .getUserInfo (); const
```

```
web3Provider
```

```
=
```

```
safeAuthPack .getProvider ();
```

```
// Subscribe to events const
```

```
handler
```

```
= (event) => {}; safeAuthPack .subscribe (packEvent , handler); safeAuthPack .unsubscribe (packEvent , handler);
```

```
// Sign out await
```

```
safeAuthPack .signOut ();
```

[AuthKitBasePack Protocol Kit](#)

Was this page helpful?

[Report issue](#)