

AuthKitBasePack

To be used as part of the Auth Kit, new packs need to extend the `AuthKitBasePack` class. This abstract class provides a common interface that subclasses must implement. It provides the specific Safe common functionality that any custom implementation can take leverage.

Install dependencies

To use the `AuthKitBasePack`, you need to install the `@safe-global/auth-kit` package.

```
yarn
add
@safe-global/auth-kit
```

Reference

This class is used to create new packs. Any new pack should extend this class and implement the abstract methods. Extending from the `AuthKitBasePack` class will give the subclass access to the common Safe features like retrieving the associated Safes for a given Ethereum address.

```
class
MyPack
extends
AuthKitBasePack { // Implementation of the abstract methods }
```

Abstract methods and properties that any pack must implement

These methods and properties are the common interface for all the Auth packs. Check each pack's documentation for more details.

isAuthenticated(): boolean

Any pack extending the `AuthKitBasePack` class must implement the `isAuthenticated()` getter. This method returns a boolean indicating whether the user is authenticated.

init(options?): Promise

Provides the initialization options for the required packages, classes and properties.

signIn(): Promise

Provides a mechanism to connect to the provider services and returns an Ethereum address that will be used as a signer along with the associated Safe addresses.

signOut(): Promise

Disconnects the provider services and cleans up any data related to the current user.

getProvider(): AuthKitEthereumProvider | null

Returns a Web3 provider that can be used to interact with the blockchain (web3.js or ethers.js for example).

getUserInfo(): userInfo

Returns the user information as a JSON object.

subscribe(event, handler)

Provides a way to subscribe to events

unsubscribe(event, handler)

Provides a way to unsubscribe from an event.

SpecificAuthKitBasePack

methods

These methods provide the functionality associated with Safe so they can be used in the implementation of the packs.

getSafes(txServiceUrl?): Promise

Returns the list of Safes associated with the signer address by internally calling the `getAddress()` method.

Parameters

- `txServiceUrl`
- (optional): The Safe Transaction Service URL to retrieve the Safes associated with the owner (signer).

Returns

- `safes[]`
- : The list of Safes associated with the owner (signer).

getAddress(): Promise

Returns the Ethereum address extracted from the provider retrieved by `getProvider()` . This returns a Safe owner (signer).

Returns

- `address`
- : The Ethereum address extracted from the provider.

getChainId(): Promise

Returns the chain ID of the connected network.

Returns

- `chainId`
- : The chain ID extracted from the provider.

[Reference SafeAuthPack](#)

Was this page helpful?

[Report issue](#)