Whitelabel Plug n Play Modal

Web3Auth is fully whitelabel with application branding to have a consistent user experience. For Whitelabeling, there are three different aspects available for you. Web3Auth gives you the ability to define a custom UI, branding, and translations for your applications. All our SDKs support Whitelabeling, giving granular customization capability across all our offerings.

note This is a paid feature and the minimum pricing plan to use this SDK in a production environment is theGrowth Plan . You can use this feature in the development environment for free. Within the Web3Auth Plug and Play Modal SDK, whitelabeling can happen in 3 different places,

- Web3Auth Modal
- : You can whitelabel the Web3Auth Modal according to your needs, for this, you need to pass on the following
- parameters:* Within theuiConfig
-
  - parameter while instantiating Web3Auth, you can configure the Web3Auth Modal according to your application's
-
  - requirements.
-
  - Change the visibility and order of methods appearing in Web3Auth modal by passing themodalConfig
-
  - ininitModal
-
  - function, while initializing the SDK.
- Openlogin Adapter
- : The redirect screens while logging in and constructing the key, can be customised according to your
- common as well. For this, you need to pass on thewhiteLabel
- configuration parameter to theopenlogin-adapter
- .
- Torus Wallet Plugin
- : The Torus Wallet Plugin can be customized according to your needs for helping you make wallet flows
- within your application. For this, you need to pass on thetorusWalletOpts
- configuration.

Web3Auth Modalâ

Whitelabeling while Web3Auth Instantiationâ

tip Read more about Instantiating Web3Auth in theInitialization SDK Reference While instantiating Web3Auth, the constructor takes an object withWeb3AuthOptions as input. TheWeb3AuthOptions object is further taken in the parameter ofuiConfig which is the configuration for whitelabeling the Modal UI display properties.

uiConfig

â

- Table
- Interface

Parameter Description loginMethodsOrder? The list of login methods can be reordered with this parameter. Those methods specified will be first on the list. Default value is ["google", "facebook", "twitter", "reddit", "discord", "twitch", "apple", "line", "github", "kakao", "linkedin", "weibo", "wechat", "email_passwordless"]. modalZIndex? Z-index of the modal and iframe. The default value is99998 and accepts astring as a value. displayErrorsOnModal? Whether to show errors on Web3Auth modal. Default value istrue . loginGridCol? Number of columns to display the Social Login buttons. Default value is 3, available options are 2 or 3. primaryButton? Decides which button will be displayed as primary button in modal. Only one button will be primary and other buttons in modal will be secondary. Default value issocialLogin . Available options areexternalLogin ,socialLogin , andemailLogin . interface

UIConfig

extends

WhiteLabelData

{ /* * *order of how login methods are shown * * @defaultValue* "google", "facebook", "twitter", "reddit", "discord", "twitch", "apple", "line", "github", "kakao", "linkedin", "weibo", "wechat", "email_passwordless"]* / loginMethodsOrder ? :

string [ ] ; /* * Z-index of the modal and iframe * @defaultValue 99998* / modalZIndex ? :

string ; /* * Whether to show errors on Web3Auth modal. * * @defaultValue*true* / displayErrorsOnModal ? :

boolean ; /* * number of columns to display the Social Login buttons * * @defaultValue*3* / loginGridCol ? :

2

|

3 ; /* * decides which button will be displayed as primary button in modal * only one button will be primary and other buttons in modal will be secondary * * @defaultValue *socialLogin* / primaryButton ? :

"externalLogin"

|

"socialLogin"

|

"emailLogin" ; }

WhitelabelData

[â](#)

whiteLabel?: WhiteLabelData;

[â](#)

The whitelabel parameter takesWhitelabelData as input. TheWhitelabelData object takes the following parameters:

- Table
- Interface

WhiteLabelData

Parameter Description appName? App name to be displayed in the User Flow Screens. It acceptsstring as a value. appUrl? App URL to be displayed in the User Flow Screens. It acceptsstring as a value. logoLight? App logo to be shown on the light background (light theme). It acceptsstring as a value. logoDark? App logo to be shown on the dark background (dark theme). It acceptsstring as a value. defaultLanguage? Default Language to use. Choose from:* en * - English * de * - German * ja * - Japanese * ko * - Korean * zh * - Mandarin * es * - Spanish * fr * - French * pt * - Portuguese * nl * - Dutch * tr * - Turkish

. Default language isen mode? Choose betweenauto ,light ordark background modes. Default isauto . theme? Used to customize the theme of the login modal with the following options 'primary' - To customize the primary color of the modal's content. It acceptsRecord as a value. tncLink? Language specific link for terms and conditions on torus-website. See (examples/vue-app) to configure e.g. tncLink:{en: "http://example.com/tnc/en", ja: "http://example.com/tnc/ja"} privacyPolicy? Language specific link for privacy policy on torus-website. See (examples/vue-app) to configure e.g.privacyPolicy: { en: "http://example.com/tnc/en", ja: "http://example.com/tnc/ja", } export

type

WhiteLabelData

=

{ /* * App name to display in the UI*/ appName ? :

string ; /* * App url */ appUrl ? :

string ; /* * App logo to use in light mode*/ logoLight ? :

string ; /* * App logo to use in dark mode*/ logoDark ? :

string ; /* * language which will be used by web3auth. app will use browser language if not specified. if language is not supported it will use "en" * en: english * de: german * ja: japanese * ko: korean * zh: mandarin * es: spanish * fr: french * pt: portuguese * nl: dutch * * @defaultValue en / defaultLanguage ? :

LANGUAGE_TYPE ; /* theme * * @defaultValue auto*/ mode ? :

THEME_MODE_TYPE ; /* * Use logo loader * * @defaultValue false/ useLogoLoader ? :

boolean ; /* * Used to customize theme of the login modal with following options *primary' - To customize primary color of modal's content. / theme ? :

{ primary ? :

string ; gray ? :

string ; red ? :

string ; green ? :

string ; success ? :

string ; warning ? :

string ; error ? :

string ; info ? :

string ; white ? :

string ; } ; /* * Language specific link for terms and conditions on torus-website. See (examples/vue-app) to configure * e.g. * tncLink: { * en: "http://example.com/tnc/en", * ja: "http://example.com/tnc/ja", * } / tncLink ? :

Partial < Record < LANGUAGE_TYPE ,

string

      ; /* * Language specific link for privacy policy on torus-website. See (examples/vue-app) to configure * e.g. * privacyPolicy: { * en: "http://example.com/tnc/en", * ja: "http://example.com/tnc/ja", * } / privacyPolicy ? :

Partial < Record < LANGUAGE_TYPE ,

string

      ; } ;

Exampleâ

const web3auth =

new

Web3Auth ({ clientId :

"" ,

// Get your Client ID from the Web3Auth Dashboard web3AuthNetwork :

WEB3AUTH_NETWORK . SAPPHIRE_MAINNET ,

// import {WEB3AUTH_NETWORK} from "@web3auth/base"; chainConfig :

{ chainNamespace :

CHAIN_NAMESPACES . EIP155 , chainId :

"0x1" , rpcTarget :

"https://rpc.ankr.com/eth" ,

// This is the mainnet RPC we have added, please pass on your own endpoint while creating an app } , uiConfig :

{ appName :

"W3A Heroes" , mode :

"dark" ,

// light, dark or auto loginMethodsOrder :

["apple" ,

"google" ,

"twitter"] , logoLight :

"https://web3auth.io/images/web3auth-logo.svg" , logoDark :

"https://web3auth.io/images/web3auth-logo---Dark.svg" , defaultLanguage :

"en" ,

// en, de, ja, ko, zh, es, fr, pt, nl, tr loginGridCol :

3 , primaryButton :

"socialLogin" ,

// "externalLogin" | "socialLogin" | "emailLogin" } , modalZIndex :

"99998" , }) ;

Whitelabeling while Modal Initialization[â](#)

tip Read more about Initializing Web3Auth in the[Initialization SDK Reference](#) While initializing the Web3Auth Modal, theinitModal() function takes an object withmodalConfig as input. This object helps you configure the different adapters available in the Web3Auth Modal. Further, themodalConfig object takes the parameter ofloginMethods which helps you configure the social logins and their visibility in the modal.

initModal()

[â](#)

modalConfig

[â](#)

- Table
- Interface

modalConfig: { ADAPTER : { params } }

Parameter Description label Label of the adapter you want to configure. It's a mandatory field which acceptsstring . showOnModal? Whether to show an adapter in modal or not. Default value istrue . showOnDesktop? Whether to show an adapter in desktop or not. Default value istrue . showOnMobile? Whether to show an adapter in mobile or not. Default value istrue . Additionally, to configure the Openlogin Adapter's each login method, we have theloginMethods? parameter.

Parameter Description loginMethods? To configure visibility of each social login method for the openlogin adapter. It acceptsLoginMethodConfig as a value. initModal (params ? :

{ modalConfig ? :

Record < WALLET_ADAPTER_TYPE ,

ModalConfig

     ; }) :

Promise < void

     ;

interface

ModalConfig

extends

BaseAdapterConfig

{ loginMethods ? :

LoginMethodConfig ; }

interface

BaseAdapterConfig

{ label :

string ; showOnModal ? :

boolean ; showOnMobile ? :

boolean ; showOnDesktop ? :

boolean ; } loginMethods: { label: { params } }

InloginMethods , you can configure the visibility of each social login method for the openlogin adapter. The social login is corresponded by thelabel parameter. Below is the table indicating the differentparams available for customization.

Forlabels you can choose between these options:google ,facebook ,twitter ,reddit ,discord ,twitch ,apple ,line ,github ,kakao ,linkedin ,weibo ,wechat ,email_passwordless

- Table
- Type Declaration

params

Parameter Description name? Display Name. It acceptsstring as a value. description? Description for the button. If provided, it renders as a full length button. else, icon button. It acceptsstring as a value. logoHover? Logo to be shown on mouse hover. It acceptsstring as a value. logoLight? Light logo for dark background. It acceptsstring as a value. logoDark? Dark logo for light background. It acceptsstring as a value. mainOption? Show login button on the main list. It acceptsoolean as a value. Default value is false. showOnModal? Whether to show the login button on modal or not. Default value is true. showOnDesktop? Whether to show the login button on desktop. Default value is true. showOnMobile? Whether to show the login button on mobile. Default value is true. declare

type

LoginMethodConfig

=

Record < string , { /* * *Display Name. If not provided, we use the default for openlogin app* name :

string ; /* * *Description for button. If provided, it renders as a full length button. else, icon button* description ? :

string ; /* * *Logo to be shown on mouse hover. If not provided, we use the default for openlogin app* logoHover ? :

string ; /* * *Logo to be shown on dark background (dark theme). If not provided, we use the default for openlogin app* logoLight ? :

string ; /* * *Logo to be shown on light background (light theme). If not provided, we use the default for openlogin app* logoDark ? :

string ; /* * *Show login button on the main list* mainOption ? :

boolean ; /* * *Whether to show the login button on modal or not* showOnModal ? :

boolean ; /* * *Whether to show the login button on desktop* showOnDesktop ? :

boolean ; /* * *Whether to show the login button on mobile* showOnMobile ? :

boolean ; }

;

Example

await web3auth . initModal ({ modalConfig :

```
{ [ WALLET_ADAPTERS . OPENLOGIN ] :
```

{ label :

"openlogin" , loginMethods :

{ google :

{ name :

"google login" , logoDark :

"url to your custom logo which will shown in dark mode" , } , facebook :

{ // it will hide the facebook option from the Web3Auth modal. name :

"facebook login" , showOnModal :

false , } , } , // setting it to false will hide all social login methods from modal. showOnModal :

true , } , } , }) ;

Openlogin Adapter[â]

Web3Auth's Social Logins and Email Login run using the OpenLogin Flow. The whole OpenLogin user experience can also be whitelabeled using OpenLogin Adapter settings. For this, you need to pass on thewhiteLabel configuration parameter to theadapterSettings property of theopenlogin-adapter .

tip Checkout theopenlogin-adapter SDK Reference for more details on different configurations you can pass for customizations.

whiteLabel?: WhiteLabelData;

[â]

The whitelabel parameter takesWhitelabelData as input. TheWhitelabelData object takes the following parameters:

- Table
- Interface

WhiteLabelData

Parameter Description appName? App name to be displayed in the User Flow Screens. It acceptsstring as a value. appUrl? App URL to be displayed in the User Flow Screens. It acceptsstring as a value. logoLight? App logo to be shown on the light background (light theme). It acceptsstring as a value. logoDark? App logo to be shown on the dark background (dark theme). It acceptsstring as a value. defaultLanguage? Default Language to use. Choose from:* en * - English * de * - German * ja * - Japanese * ko * - Korean * zh * - Mandarin * es * - Spanish * fr * - French * pt * - Portuguese * nl * - Dutch * tr * - Turkish

. Default language isen mode? Choose betweenauto ,light ordark background modes. Default isauto . theme? Used to customize the theme of the login modal with the following options 'primary' - To customize the primary color of the modal's content. It acceptsRecord as a value. tncLink? Language specific link for terms and conditions on torus-website. See (examples/vue-app) to configure e.g. tncLink:{en: "http://example.com/tnc/en", ja: "http://example.com/tnc/ja"} privacyPolicy? Language specific link for privacy policy on torus-website. See (examples/vue-app) to configure e.g.privacyPolicy: { en: "http://example.com/tnc/en", ja: "http://example.com/tnc/ja", } export

type

WhiteLabelData

=

{ /* * App name to display in the UI*/ appName ? :

string ; /* * App url*/ appUrl ? :

string ; /* * App logo to use in light mode*/ logoLight ? :

string ; /* * App logo to use in dark mode*/ logoDark ? :

string ; /* * language which will be used by web3auth. app will use browser language if not specified. if language is not

supported it will use "en" * en: english * de: german * ja: japanese * ko: korean * zh: mandarin * es: spanish * fr: french * pt: portuguese * nl: dutch * * @defaultValue en / defaultLanguage ? :

LANGUAGE_TYPE ; /* theme * * @defaultValue auto/ mode ? :

THEME_MODE_TYPE ; /* * Use logo loader * * @defaultValue false/ useLogoLoader ? :

boolean ; /* * Used to customize theme of the login modal with following options *'primary' - To customize primary color of modal's content. / theme ? :

{ primary ? :

string ; gray ? :

string ; red ? :

string ; green ? :

string ; success ? :

string ; warning ? :

string ; error ? :

string ; info ? :

string ; white ? :

string ; } ; /* * Language specific link for terms and conditions on torus-website. See (examples/vue-app) to configure * e.g. * tncLink: { * en: "http://example.com/tnc/en", * ja: "http://example.com/tnc/ja", * } / tncLink ? :

Partial < Record < LANGUAGE_TYPE ,

string

     ; /* * Language specific link for privacy policy on torus-website. See (examples/vue-app) to configure * e.g. * privacyPolicy: { * en: "http://example.com/tnc/en", * ja: "http://example.com/tnc/ja", * } / privacyPolicy ? :

Partial < Record < LANGUAGE_TYPE ,

string

     ; } ;

name

[â](#)

The name of the application. This will be displayed in the key reconstruction page.

Standard screenwithout

any change

Name changed toFormidable Duo

logoLight

&logoDark [â](#)

The logo of the application. Displayed in dark and light mode respectively. This will be displayed in the key reconstruction page.

logoLight

on dark mode

logoDark

on light mode

defaultLanguage

[â](#)

Default language will set the language used on all OpenLogin screens. The supported languages are:

- en
-
  - English (default)
- de
-
  - German
- ja
-
  - Japanese
- ko
-
  - Korean
- zh
-
  - Mandarin
- es
-
  - Spanish
- fr
-
  - French
- pt
-
  - Portuguese
- nl
-
  - Dutch

dark

[â](#)

Can be set totrue orfalse with default set tofalse .

For Light:dark: false

For Dark:dark: true

theme

[â](#)

Theme is a record of colors that can be configured. As of, now onlyprimary color can be set and has effect on OpenLogin screens (default primary color is#0364FF). Theme affects icons and links. Examples below.

Standard color#0364FF

Color changed to#D72F7A

Example[â](#)

import

{

OpenloginAdapter

}

from

```
"@web3auth/openlogin-adapter" ;

const openloginAdapter =

new

OpenloginAdapter ( { adapterSettings :

{ clientId ,

//Optional - Provide only if you haven't provided it in the Web3Auth Instantiation Code network :

"sapphire_mainnet" ,

// Optional - Provide only if you haven't provided it in the Web3Auth Instantiation Code uxMode :

"popup" , whiteLabel :

{ appName :

"W3A Heroes" , appUrl :

"https://web3auth.io" , logoLight :

"https://web3auth.io/images/web3auth-logo.svg" , logoDark :

"https://web3auth.io/images/web3auth-logo---Dark.svg" , defaultLanguage :

"en" ,

// en, de, ja, ko, zh, es, fr, pt, nl, tr mode :

"dark" ,

// whether to enable dark mode. defaultValue: auto theme :

{ primary :

"#00D1B2" , } , useLogoLoader :

true , } , } , privateKeyProvider , } ) ; web3auth . configureAdapter ( openloginAdapter ) ;
```

Torus Wallet Plugin[â](#)

The Torus Wallet Plugins help you get the functionalities of the Torus Wallets into your Web3Auth Instance. They help you get the UI of the Torus Wallet Embedded Interface into your dApp, giving your user an overview of their wallet.

tip Read more about Initializing the Torus EVM Wallet Plugin in the [Torus Wallet Connector Plugin SDK Reference](#) While initializing the Wallet Services plugin, you need to pass on the following parameters to the constructor:

- Table
- Interface

Parameter Description wsEmbedOpts? Configuration options for Wallet Services. It acceptsPartialas a value. walletInitOptions? Parameters to customize your Wallet Services UI within the application. It acceptsPartial as a value. constructor(options?: { wsEmbedOpts?: Partial; walletInitOptions?: Partial; });

1.wsEmbedOpts

[â](#)

wsEmbedOpts is an optional parameter that allows you to pass in the configuration options for Wallet Services. It takes theCtorArgs object as input which contains the following parameters:

- Table
- Interface

CtorArgs

Parameter Description modalZIndex? Z-index of the modal and iframe. Default value is 99999 web3AuthClientId Web3Auth Client ID instring . web3AuthNetwork Web3auth network to be used. It acceptsOPENLOGIN_NETWORK_TYPE as a value. export

interface

CtorArgs

{ /* * Z-index of the modal and iframe * @defaultValue 99999*/ modalZIndex ? :

number ; /* * You can get your clientId/projectId by registering your * dapp on {@link "https://dashboard.web3auth.io"|
developer dashboard} / web3AuthClientId :

string ; /* * network specifies the web3auth network to be used./ web3AuthNetwork :

OPENLOGIN_NETWORK_TYPE ; }

export

type

OPENLOGIN_NETWORK_TYPE

=

(typeof

OPENLOGIN_NETWORK) [keyof

typeof

OPENLOGIN_NETWORK] ; export

declare

const

OPENLOGIN_NETWORK :

{ readonly

MAINNET :

"mainnet" ; readonly

TESTNET :

"testnet" ; readonly

CYAN :

"cyan" ; readonly

AQUA :

"aqua" ; readonly

CELESTE :

"celeste" ; readonly

SAPPHIRE_DEVNET :

"sapphire_devnet" ; readonly

SAPPHIRE_MAINNET :

"sapphire_mainnet" ; } ;

2.walletInitOptions

[â](#)

ThewalletInitOptions from Wallet Services is an optional parameter that allows you to customise your Wallet Services UI
within the application. It takes the objectWsEmbedParams as input which contains multiple parameters that allow you to
customise the UI of the plugin.

- Table
- Interface

WsEmbedParams

Parameter Description chainConfig? Chain config of the Blockchain to connect with. It acceptsEthereumProviderConfig . whiteLabel? White Label parameters to whitelisting the Wallet Services UI. It acceptsWhiteLabelParams . export

interface

WsEmbedParams

{ /* * Chain to connect with/ chainConfig ? :

EthereumProviderConfig ; /* * Build Environment of WsEmbed. * * production uses https://wallet.web3auth.io, * * staging uses https://staging-wallet.web3auth.io, * * testing uses https://develop-wallet.web3auth.io (latest internal build) * * development uses http://localhost:4050 (expects wallet-services-frontend to be run locally), * * @defaultValue production / buildEnv ? :

WS_EMBED_BUILD_ENV_TYPE ; /* * Enables or disables logging. * * Defaults to false in prod and true in other environments / enableLogging ? :

boolean ; /* * url of widget to load * * Defaults to true * @defaultValue true/ walletUrls ? :

Partial < Record < WS_EMBED_BUILD_ENV_TYPE ,

WalletUrlConfig

; / * **Determines how to show confirmation screens * * @defaultValue none for wallet services * @defaultValue popup for safe-auth, mocaverse *** / / * Allows you to customize the look & feel of the widget */ whiteLabel ? :

{ /* * whether to show/hide ws-embed widget. * * Defaults to true * @defaultValue true/ showWidgetButton ? :

boolean ; /* * Determines where the wsEmbed widget is visible on the page. * @defaultValue bottom-left/ buttonPosition ? :

BUTTON_POSITION_TYPE ; hideNftDisplay ? :

boolean ; hideTokenDisplay ? :

boolean ; hideTransfers ? :

boolean ; hideTopup ? :

boolean ; hideReceive ? :

boolean ; defaultPortfolio ? :

"token"

|

"nft" ; }

&

WhiteLabelData ; } ThewhiteLabel parameter takes some basic parameters andWhiteLabelParams object as input which contains the following parameters:

- Table
- Interface

Parameter Description showWidgetButton? Whether to show/hide the Wallet Service widget button. Default value istrue . buttonPosition? Determines where the Wallet Service widget is visible on the page. [bottom-left ,bottom-right ,top-left ,top-right]. Default isbottom-left . hideNftDisplay? Whether to hide the NFT display screen or not. hideTokenDisplay? Whether to hide the Token display screen or not. hideTransfers? Whether to hide the Transfer button and screen or not. hideTopup? Whether to hide the Topup button and screen or not. hideReceive? Whether to hide the Receive button and screen or not. defaultPortfolio? The default portfolio to be displayed on the home display screen. Available options are [token ,nft]. appName? App name to be displayed in the User Flow Screens. It acceptsstring as a value. appUrl? App URL to be displayed in the User Flow Screens. It acceptsstring as a value. logoLight? App logo to be shown on the light background (light theme). It acceptsstring as a value. logoDark? App logo to be shown on the dark background (dark theme). It

acceptsstring as a value. defaultLanguage? Default Language to use. Choose from:* en * - English * de * - German * ja * - Japanese * ko * - Korean * zh * - Mandarin * es * - Spanish * fr * - French * pt * - Portuguese * nl * - Dutch * tr * - Turkish

. Default language isen mode? Choose betweenauto ,light ordark background modes. Default isauto . theme? Used to customize the theme of the login modal with the following options 'primary' - To customize the primary color of the modal's content. It acceptsRecord as a value. tncLink? Language specific link for terms and conditions on torus-website. See (examples/vue-app) to configure e.g. tncLink:{en: "http://example.com/tnc/en", ja: "http://example.com/tnc/ja"} privacyPolicy? Language specific link for privacy policy on torus-website. See (examples/vue-app) to configure e.g.privacyPolicy: { en: "http://example.com/tnc/en", ja: "http://example.com/tnc/ja", } whiteLabel ? :

{ /* * whether to show/hide ws-embed widget. * * Defaults to true * @defaultValue true*/ showWidgetButton ? :

boolean ; /* * Determines where the wsEmbed widget is visible on the page. * @defaultValue bottom-left* buttonPosition ? :

BUTTON_POSITION_TYPE ; hideNftDisplay ? :

boolean ; hideTokenDisplay ? :

boolean ; hideTransfers ? :

boolean ; hideTopup ? :

boolean ; hideReceive ? :

boolean ; defaultPortfolio ? :

"token"

|

"nft" ; }

&

WhiteLabelData ;

export

declare

const

BUTTON_POSITION :

{ readonly

BOTTOM_LEFT :

"bottom-left" ; readonly

TOP_LEFT :

"top-left" ; readonly

BOTTOM_RIGHT :

"bottom-right" ; readonly

TOP_RIGHT :

"top-right" ; } ; export

type

BUTTON_POSITION_TYPE

=

(typeof

BUTTON_POSITION) [keyof

typeof

BUTTON_POSITION] ;

export

type

WhiteLabelData

=

{ /* * App name to display in the UI/ appName ? :

string ; /* * App url / appUrl ? :

string ; /* * App logo to use in light mode/ logoLight ? :

string ; /* * App logo to use in dark mode/ logoDark ? :

string ; /* * language which will be used by web3auth. app will use browser language if not specified. if language is not supported it will use "en" * en: english * de: german * ja: japanese * ko: korean * zh: mandarin * es: spanish * fr: french * pt: portuguese * nl: dutch * tr: turkish * * @defaultValue en / defaultLanguage ? :

LANGUAGE_TYPE ; /* theme * * @defaultValue auto/ mode ? :

THEME_MODE_TYPE ; /* * Use logo loader * * @defaultValue false/ useLogoLoader ? :

boolean ; /* * Used to customize theme of the login modal with following options *'primary' - To customize primary color of modal's content. / theme ? :

{ primary ? :

string ; gray ? :

string ; red ? :

string ; green ? :

string ; success ? :

string ; warning ? :

string ; error ? :

string ; info ? :

string ; white ? :

string ; } ; /* * Language specific link for terms and conditions on torus-website. See (examples/vue-app) to configure * e.g. * tncLink: { * en: "http://example.com/tnc/en", * ja: "http://example.com/tnc/ja", * } / tncLink ? :

Partial < Record < LANGUAGE_TYPE ,

string

; /* * Language specific link for privacy policy on torus-website. See (examples/vue-app) to configure * e.g. * privacyPolicy: { * en: "http://example.com/tnc/en", * ja: "http://example.com/tnc/ja", * } / privacyPolicy ? :

Partial < Record < LANGUAGE_TYPE ,

string

; } ; ;

Exampleâ

In this example, we're adding theWalletServicesPlugin with a very basic configuration.

import

{

```
WalletServicesPlugin

}

from

"@web3auth/wallet-services-plugin" ;

const walletServicesPlugin =

new

WalletServicesPlugin ( ) ; web3auth . addPlugin ( walletServicesPlugin ) ;

// Show the wallet services plugin await walletServicesPlugin . showWalletUi ( ) ;

// Show the wallet connect scanner await walletServicesPlugin . showWalletConnectScanner ( ) ;

// Show the checkout await walletServicesPlugin . showCheckout ( ) ;
```