

# Whitelabel PnP Web No Modal SDK

Web3Auth is fully whitelabelable with application branding to have a consistent user experience. All our SDKs support Whitelabeling, giving granular customization capability across all our offerings. The major benefit of the Plug and Play No Modal SDK is the flexibility in how you can make your own UI for triggering the login. Further, Whitelabeling enables you to customize the other aspects as well.

note This is a paid feature and the minimum [pricing plan](#) to use this SDK in a production environment is the Growth Plan . You can use this feature in the development environment for free. Within the Web3Auth Plug and Play No Modal SDK, whitelabeling can happen in 2 different places,

- [Openlogin Adapter](#)
- : The redirect screens while logging in and constructing the key, can be customized according to your common as well. For this, you need to pass on the whiteLabel configuration parameter to the [openlogin-adapter](#)
- .
- [uiConfig](#)
- : The UI components can be customized using the uiConfig parameter while initializing the Web3AuthNoModal SDK.

**whiteLabel?: WhiteLabelData;**

[^](#)

The whiteLabel parameter takes WhiteLabelData as input. The WhiteLabelData object takes the following parameters:

- Table
- Interface

WhiteLabelData

Parameter Description appName? App name to be displayed in the User Flow Screens. It accepts string as a value. appUrl? App URL to be displayed in the User Flow Screens. It accepts string as a value. logoLight? App logo to be shown on the light background (light theme). It accepts string as a value. logoDark? App logo to be shown on the dark background (dark theme). It accepts string 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 is en mode? Choose between auto, light or dark background modes. Default is auto . 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 accepts Record 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

[^](#)

Can be set to true or false with default set to false .

**For Light:dark: false**

**For Dark:dark: true**

## theme

[^](#)

Theme is a record of colors that can be configured. As of, now only primary 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**

## Openlogin Adapter [^](#)

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 the whiteLabel configuration parameter to the adapterSettings property of the [openlogin-adapter](#) .

tip Checkout the [openlogin-adapter](#) SDK Reference for more details on different configurations you can pass for customizations.

## Example [^](#)

```
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 ) ;

```

## uiConfig

The UI components can be customized using the `uiConfig` parameter while initializing the `Web3AuthNoModal` SDK. The `uiConfig` parameter is a `WhiteLabelData` object which can be used to customize the UI components.

### Example

```

import
{
Web3AuthNoModal
}
from
"@web3auth/no-modal" ; import
{
WhiteLabelData
}
from

```

```

"@web3auth/base" ;

const uiConfig :
WhiteLabelData

=

{ appName :
"My App" , appURL :
"https://example.com" , logoLight :
"https://example.com/logo-light.png" , logoDark :
"https://example.com/logo-dark.png" , defaultLanguage :
"en" , mode :
"light" , useLogoLoader :
true , theme :
{ primary :
"#FF0000" , secondary :
"#00FF00" , warning :
"#FFA500" , } , tncLink :
"https://example.com/terms" , privacyPolicy :
"https://example.com/privacy" , } ;

const web3auth =
new
Web3AuthNoModal ( { clientId :
"" ,

// Get your Client ID from the Web3Auth Dashboard chainConfig , web3AuthNetwork :
"sapphire_mainnet" , privateKeyProvider , uiConfig , } ) Edit this page Previous Usage Next Custom Authentication

```