Whitelabel PnP Unreal Engine SDK

For defining custom UI, branding, and translations for your brand app, you just need to specify an additional parameter within theWeb3AuthOptions class calledwhiteLabel . This parameter takes another object calledWhiteLabelData .

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.

Arguments[â](#)

WhiteLabelData

[â](#)

- Table
- Interface

Parameter Description appName? Display name for the app in the UI. logoLight? App logo to be used in dark mode. It accepts url inFString as a value. logoDark? App logo to be used in light mode. It accepts url inFString as a value. defaultLanguage? Language which will be used by Web3Auth, app will use browser language if not specified. Default language isFLanguage::en . CheckoutFLanguage for supported languages. mode? Theme mode for the login modal. Choose betweenFThemeModes::auto ,FThemeModes::light orFThemeModes::dark background modes. theme? Used to customize the theme of the login modal. It acceptsTMap as a value. appUrl? Url to be used in the Modal. It accepts url inFString as a value. useLogoLoader? Use logo loader. IflogoDark andlogoLight are null, the default Web3Auth logo will be used for the loader. Default value is false. USTRUCT (BlueprintType) struct

FWhiteLabelData { GENERATED_BODY ( )

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString appName ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString logoLight ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString logoDark ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FLanguage defaultLanguage = FLanguage :: en ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FThemeModes mode ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) TMap < FString , FString

theme ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString appUrl ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) bool useLogoLoader ;

FWhiteLabelData ( )

{ } ;

void

operator

(const FWhiteLabelData & other)

{ appName = other . appName ; logoLight = other . logoLight ; logoDark = other . logoDark ; defaultLanguage = other . defaultLanguage ; mode = other . mode ; theme = other . theme ; appUrl = other . appUrl ; useLogoLoader = other . useLogoLoader ; }

} ;

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](#)

[â](#)