

Initializing PnP Flutter SDK

After installation, the next step to use Web3Auth is to initialize the SDK. We can initialize the SDK using `init` and `initialize` function.

Please note that these are the most critical steps where you need to pass on different parameters according to the preference of your project. Additionally, Whitelabeling and Custom Authentication have to be configured within this step, if you wish to customize your Web3Auth Instance.

Using the `init`

function [@](#)

Initialize the `Web3AuthFlutter` plugin and configure the `whiteLabel` and custom authentication as per your use.

Future < void

`initWeb3Auth ()`

`async`

`{`

`late final`

`Uri redirectUrl ;`

`if`

`(Platform . isAndroid)`

`{ redirectUrl =`

`Uri . parse ('{SCHEME}://{HOST}/auth') ; // w3a://com.example.w3aflutter/auth }`

`else`

`{ redirectUrl =`

`Uri . parse ('{bundleId}://auth') ; // com.example.w3aflutter://auth }`

`await`

`Web3AuthFlutter . init (Web3AuthOptions (clientId :`

`"WEB3AUTH_CLIENT_ID" , network :`

`Network . sapphire_mainnet , redirectUrl : redirectUrl ,)) ;`

`// Just after init method, invoke the initialize function await`

`Web3AuthFlutter . initialize () ; }`

Web3AuthOptions [@](#)

The `Web3AuthFlutter` takes an object called `Web3AuthOptions` as input. Below are the available fields of the `Web3AuthOptions` class.

- Table
- Class

Parameter Description
`clientId` Your Web3Auth Client ID. You can get it from Web3Auth [Dashboard](#) under project details. It's a mandatory field of type `String`
`network` Web3Auth Network, `sapphire_mainnet` , `sapphire_devnet` , `mainnet` , `cyan` , `aqua` , `testnet` . It's a mandatory field of type [Network](#) .
`redirectUrl` URL that Web3Auth will redirect API responses upon successful authentication from browser. It's a mandatory field of type [Uri](#) .
`whiteLabel?` WhiteLabel options for web3auth. It helps you define custom UI, branding, and translations for your brand app. It takes [WhiteLabelData](#) as a value.
`loginConfig?` Login config for the custom verifiers. It takes `HashMap` as a value.
`useCoreKitKey?` Use CoreKit Key to get core kit key. It's an optional field with default value as `false` .
`chainNamespace?` Chain Namespace [`eip155` and `solana`]. It takes [ChainNamespace](#) as a value.
`mfaSettings?` Allows developers to configure the Mfa settings for authentication. It takes [MfaSettings](#) as a value.
`sessionTime?` It allows developers to configure the session management time. Session Time is

in seconds, default is 86400 seconds which is 1 day.sessionTime can be max 7 days class

Web3AuthOptions

```
{ final
String clientId ; final
Network network ; final
BuildEnv ? buildEnv ; final
String ? sdkUrl ; final
Uri redirectUrl ; final
WhiteLabelData ? whiteLabel ; final
HashMap < String ,
LoginConfigItem
    ? loginConfig ; final bool ? useCoreKitKey ; final
ChainNamespace ? chainNamespace ; final
MfaSettings ? mfaSettings ; final int ? sessionTime ;
Web3AuthOptions ( { required this . clientId , required this . network , this . buildEnv =
BuildEnv . production , String ? sdkUrl , required this . redirectUrl , this . whiteLabel , this . loginConfig , this . useCoreKitKey
, this . chainNamespace =
ChainNamespace . eip155 , this . sessionTime =
86400 , this . mfaSettings , } ) : sdkUrl = sdkUrl ? ?
getSdkUrl ( buildEnv ? ?
BuildEnv . production ) ;
Map < String ,
dynamic
toJson ( )
{ return
{ 'clientId' : clientId , 'network' : network . name , 'sdkUrl' : sdkUrl , 'buildEnv' : buildEnv ? . name , 'redirectUrl' : redirectUrl .
toString ( ) , 'whiteLabel' : whiteLabel ? . toJson ( ) , 'loginConfig' : loginConfig , 'useCoreKitKey' : useCoreKitKey ,
'chainNamespace' : chainNamespace ? . name , 'mfaSettings' : mfaSettings , "sessionTime" : sessionTime , } ; } } Edit this
page Previous Install Next Usage
```