Initializing PnP Unity SDK

After installation, the next step in using Web3Auth is to initialize the SDK.

However, initialization is a two-step process:

1. [Creating a Web3Auth instance](#)
2. [Setting up Web3Auth options](#)

Please note that these are the most critical steps where you will need to pass different parameters according to your project's preference. Additionally, you must configure whitelabeling and custom authentication within this step if you want to customize your Web3Auth instance.

Create Web3Auth Instance[â](#)

Attach aWeb3Auth.cs script to your game object where you want to write your authentication code.

tip You can refer to following sample file on how your boilerplate script should look like:

/Assets/Web3Auth.cs using

System ; using

System . Linq ; using

System . Collections . Generic ; using

UnityEngine ; using

UnityEngine . UI ; using

Newtonsoft . Json ;

public

class

Web3Auth

:

MonoBehaviour { // Start is called before the first frame update void

Start ( )

{ } public

void

login ( )

{ } private

void

onLogin (Web3AuthResponse response)

{ } public

void

logout ( )

{ } private

void

onLogout ( )

{ } } Within your script, import theWeb3Auth component in your class.

Web3Auth web3Auth ; Next, you need to create an instance within yourStart() function by creating an instance of the

component you just imported.

web3Auth

GetComponent < Web3Auth

( ) ;

Setting up Web3Auth Options[â]

After instantiation, within yourStart() function, set up the Web3Auth Options as follows:

web3Auth . setOptions (new

Web3AuthOptions ( ) {

}) ;

Arguments[â]

Web3AuthOptions

[â]

The Web3Auth Constructor takes a classWeb3AuthOptions as input. This class has the following arguments.

- Table
- Interface

Parameter Description clientId Your Web3Auth Client ID. You can get it from the Web3AuthDashboard under project details. It's a mandatory field of typestring network Defines the Web3Auth network. 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 typeUri . whiteLabel? WhiteLabel options for web3auth. It helps you define custom UI, branding, and translations for your brand app. It takesWhiteLabelData as a value. loginConfig? Login config for the custom verifiers. It takesDictionary as a value. useCoreKitKey? Use CoreKit Key to get core kit key. It's an optional field with default value asfalse . chainNamespace? Chain Namespace [EIP155 andSOLANA]. It takesWeb3Auth.ChainNamespace as a value. mfaSettings? Allows developers to configure the MFA settings for authentication. It takesMfaSettings as a value. class

Web3AuthOptions

{ string clientId ;

// Your Web3Auth project ID public

Web3Auth . Network network ;

// Network to run Web3Auth, either SAPPHIRE_MAINNET, SAPPHIRE_DEVNET, MAINNET, TESTNET, AQUA or CYAN public

Uri redirectUrl ;

// URL that Web3Auth will redirect API responses public

WhiteLabelData ? whiteLabel ;

// Optional param to configure look public

Dictionary < string , LoginConfigItem

loginConfig ;

// Optional public

bool ? useCoreKitKey ;

// Optional public

Web3Auth . ChainNamespace ? chainNamespace ;

// Optional, default is "EIP155" public

MfaSettings ? mfaSettings ;

// Optional } tip You can also configure yourclient_id ,redirect_url andnetwork within the script settings in the Unity Editor. It will look something like this:

Example[â]

```
using

System ; using

System . Linq ; using

System . Collections . Generic ; using

UnityEngine ; using

UnityEngine . UI ; using

Newtonsoft . Json ;

public

class

Web3custom

:

MonoBehaviour { Web3Auth web3Auth ;

// Start is called before the first frame update void

Start ( ) { web3Auth =

GetComponent < Web3Auth

( ) ; web3Auth . setOptions ( new

Web3AuthOptions ( ) { redirectUrl =

new

Uri ( "torusapp://com.torus.Web3AuthUnity/auth" ) , clientId =

"BAwFgL-r7wzQKmtcdiz2uHJKNZdK7gzEf2q-m55xfzSZOw8jLOyIi4AVvvzaEQO5nv2dFLEmf9LBkF8kaq3aErg" , network =
Web3Auth . Network . TESTNET , } ) ; web3Auth . onLogin += onLogin ; web3Auth . onLogout += onLogout ; } public

void

login ( )

{ } private

void

onLogin ( Web3AuthResponse response )

{ } public

void

logout ( )

{ } private

void

onLogout ( )

{ } }
```