

Initializing PnP Unity SDK

After Installation, the next step to 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 must pass on different parameters according to the preference of your project. Additionally, You must configure Whitelabeling and Custom Authentication within this step if you wish to customize your Web3Auth Instance.

Create Web3Auth Instance

Attach aWeb3Auth.cs script to your game object where you are planning 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 create an instance within yourStart() function by creating an instance of the component you just

imported.

web3Auth

```
GetComponent < Web3Auth  
    () ;
```

Setting up Web3Auth Options

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

```
web3Auth . setOptions ( new  
Web3AuthOptions ( ) {  
}  
} ) ;
```

Arguments

Web3AuthOptions

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 Web3Auth [Dashboard](#) under project details. It's a mandatory field of type string
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 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 Dictionary 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 Web3Auth.ChainNamespace as a value.
mfaSettings? Allows developers to configure the Mfa settings for authentication. It takes MfaSettings 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 setting in Unity Editor. It will look something like this:

Example

```
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-m55xfzSZOw8jLOyli4AVvvzaEQO5nv2dFLEmf9LBkF8kaq3aErg" , network =
Web3Auth . Network . TESTNET , } ) ; web3Auth . onLogin += onLogin ; web3Auth . onLogout += onLogout ; } public
void
login ( )
{ } private
void
onLogin ( Web3AuthResponse response )
{ } public
void
logout ( )
{ } private
void
onLogout ( )
{ } } Edit this page Previous Install Next Usage
```