

Multi Factor Authentication in PnP iOS SDK

The Multi Factor Authentication feature refers to the ability of the user to create a backup share (recovery phrase). This helps them login into a new device and also to recover their account if they lose their original device.

Minimum Growth plan required 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. You can set the mfaLevel within the loginParams to customize when mfa screen should be shown to user. It currently accepts 4 values:

- default
 - Setting the mfaLevel to default will present the MFA screen to user on every third login.mfaLevel = MFALevel.DEFAULT
- optional
 - Setting mfaLevel to optional will present the MFA screen to user on every login but user will have the option to skip it.mfaLevel = MFALevel.OPTIONAL
- mandatory
 - Setting mfaLevel to mandatory will make it mandatory for user to setup MFA after login.mfaLevel = MFALevel.MANDATORY
- none
 - Setting mfaLevel to none will skip the mfa setup screen totally.mfaLevel = MFALevel.NONE

Note If you are using default verifiers, your users may have set up MFA on other dApps that also use default Web3Auth verifiers. In this case, the MFA screen will continue to appear if the user has enabled MFA on other dApps. This is because MFA cannot be turned off once it is enabled. Web3Auth () . login (W3ALoginParams (loginProvider : provider , mfaLevel =

MFALevel . MANDATORY)) Usage import

Foundation import

Web3Auth

class

ViewModel :

ObservableObject

{ var web3Auth :

Web3Auth ? @Published

var loggedIn :

Bool

=

false @Published

var user :

Web3AuthState ? @Published

var isLoading =

false @Published

var navigationTitle :

String

=

""

func

```

setup ( )
async
{ guard web3Auth ==
nil
else
{
return
} await
MainActor . run ( body :
{ isLoading =
true navigationTitle =
>Loading" } ) web3Auth =
await
Web3Auth ( W3AInitParams ( clientId : clientId , network : network ) ) await
MainActor . run ( body :
{ if
self . web3Auth ? . state !=
nil
{ user = web3Auth ? . state loggedIn =
true } isLoading =
false navigationTitle = loggedIn ?
>UserInfo"
:
>SignIn" } ) }
func
login ( provider :
Web3AuthProvider )
{ Task
{ do
{ let result =
try
await web3Auth . login ( W3ALoginParams ( // provider can be .GOOGLE, .FACEBOOK, .APPLE etc loginProvider : provider
, mfaLevel :
MFALevel . MANDATORY ) ) await
MainActor . run ( body :
{ user = result loggedIn =
true } )
}

```

```
catch
```

```
{ print ( "Error" ) } } }
```

Using themfaSettings

to configure MFA Order

Minimum SCALE plan required This is a paid feature and the minimum [pricing plan](#) to use this SDK in a production environment is the SCALE Plan . You can use this feature in the development environment for free. You can configure the order of MFA or enable/disable MFA type by passing themfaSettings object in Web3AuthOptions .

MfaSettings

- Table
- Class

Parameter Description deviceShareFactor? MFA setting for deviceShareFactor. It acceptsMfaSetting as a value. backUpShareFactor? MFA setting for backUpShareFactor. It acceptsMfaSetting as a value. socialBackupFactor? MFA setting for socialBackupFactor. It acceptsMfaSetting as a value. passwordFactor? MFA setting for passwordFactor. It acceptsMfaSetting as a value. public

```
struct
```

```
MfaSettings :
```

```
Codable
```

```
{ public
```

```
init ( deviceShareFactor :
```

```
MfaSetting ? , backUpShareFactor :
```

```
MfaSetting ? , socialBackupFactor :
```

```
MfaSetting ? , passwordFactor :
```

```
MfaSetting ? )
```

```
{ self . deviceShareFactor = deviceShareFactor self . backUpShareFactor = backUpShareFactor self . socialBackupFactor = socialBackupFactor self . passwordFactor = passwordFactor }
```

```
let deviceShareFactor :
```

```
MfaSetting ? let backUpShareFactor :
```

```
MfaSetting ? let socialBackupFactor :
```

```
MfaSetting ? let passwordFactor :
```

```
MfaSetting ? } MfaSettings
```

- Table
- Class

Parameter Description enable Enable/Disable MFA. It acceptsBool as a value. priority? Priority of MFA. It acceptsInt as a value, where valid range is from 1 to 4. mandatory? Mandatory/Optional MFA. It acceptsBool as a value. public

```
struct
```

```
MfaSetting :
```

```
Codable
```

```
{ public
```

```
init ( enable :
```

```
Bool , priority :
```

```
Int ? , mandatory :
```

```

Bool ?
=
nil )
{ self . enable = enable self . priority = priority self . mandatory = mandatory }
let enable :
Bool let priority :
Int ? let mandatory :
Bool ? } let result =
try
await
Web3Auth ( W3AInitParams ( clientId :
"YOUR_CLIENT_ID" , network :
. sapphire_devnet , loginConfig :
[ TypeOfLogin . google . rawValue :
. init ( // Get it from Web3Auth Dashboard verifier :
"YOUR_VERIFIER_ID" , typeOfLogin :
. google , name :
"Web3Auth-Aggregate-Verifier-Google-Example" , clientId :
"YOUR_GOOGLE_CLIENT_ID" , // Get it from Web3Auth Dashboard verifierSubIdentifier :
"YOUR_VERIFIER_SUB_ID" ) ] , whiteLabel :
W3AWhiteLabelData ( appName :
"Web3Auth Stub" , logoLight :
"https://images.web3auth.io/web3auth-logo-w.svg" , logoDark :
"https://images.web3auth.io/web3auth-logo-w.svg" , defaultLanguage :
. en ,
// en, de, ja, ko, zh, es, fr, pt, nl mode :
. dark , theme :
[ "primary" :
"#d53f8c" ] ) , mfaSettings :
MfaSettings ( deviceShareFactor :
MfaSetting ( enable :
true , priority :
1 ) , backUpShareFactor :
MfaSetting ( enable :
true , priority :
2 ) , socialBackupFactor :
MfaSetting ( enable :

```

```
true , priority :  
3 ) , passwordFactor :  
MfaSetting ( enable :  
true , priority :  
4 ) ) ) . login ( W3ALoginParams ( loginProvider :  
. GOOGLE , dappShare :  
nil , extraLoginOptions :  
ExtraLoginOptions ( display :  
nil , prompt :  
nil , max_age :  
nil , ui_locales :  
nil , id_token_hint :  
nil , id_token :  
nil , login_hint :  
nil , acr_values :  
nil , scope :  
nil , audience :  
nil , connection :  
nil , domain :  
nil , client_id :  
nil , redirect_uri :  
nil , leeway :  
nil , verifierIdField :  
nil , isVerifierIdCaseSensitive :  
nil , additionalParams :  
nil ) , mfaLevel :  
. DEFAULT , curve :  
. SECP256K1 ) ) Edit this page Previous Custom Authentication Next dApp Share
```