

# Multi Factor Authentication in PnP React Native SDK

The Multi-Factor Authentication feature refers to the ability of the user to create a backup share (recovery phrase). This helps them log 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 `themfaLevel` within `web3auth.login()` to customize when the mfa screen should be shown to the user. It currently accepts 4 values:

- default
  - Setting the `mfaLevel` to default will present the MFA screen to the user on every third login. `mfaLevel = MFALevel.DEFAULT`
- optional
  - Setting `mfaLevel` to optional will present the MFA screen to the user on every login but the user will have the option to skip it. `mfaLevel = MFALevel.OPTIONAL`
- mandatory
  - Setting `mfaLevel` to mandatory will make it mandatory for the user to set up 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.

```
const
```

```
MFA_LEVELS
```

```
=
```

```
{ DEFAULT :
```

```
"default" , OPTIONAL :
```

```
"optional" , MANDATORY :
```

```
"mandatory" , NONE :
```

```
"none" , } ;
```

## Using `themfaSettings`

to configure the MFA Order [a](#)

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 `OpenLoginOptions` . We offer the following backup factors under `mfaSettings` :

- `deviceShareFactor`
- ,
- `backUpShareFactor`
- ,
- `socialBackupFactor`
- , and
- `passwordFactor`
- .

```
export
```

```
declare
```

```
const
```

```

MFA_FACTOR :
{ readonly
DEVICE :
"deviceShareFactor" ; readonly
BACKUP_SHARE :
"backUpShareFactor" ; readonly
SOCIAL_BACKUP :
"socialBackupFactor" ; readonly
PASSWORD :
"passwordFactor" ; } ; export
type
MFA_FACTOR_TYPE
=
( typeof
MFA_FACTOR ) [ keyof
typeof
MFA_FACTOR ] ; export
type
MFA_SETTINGS
=
{ enable :
boolean ; priority ? :
number ; mandatory ? :
boolean ; } ; export
type
MfaSettings
=
Partial < Record < MFA_FACTOR_TYPE ,
MFA_SETTINGS
;

```

## Example[a](#)

```

const state =
await web3auth . login ( { loginProvider :
LoginProvider . GOOGLE , redirectUrl : resolvedRedirectUrl , mfaLevel :
MFALevel . MANDATORY ,
// MFALevel.DEFAULT, MFALevel.OPTIONAL, MFALevel.MANDATORY, MFALevel.NONE // SCALE and above plan only
feature mfaSettings :

```

```
{ deviceShareFactor :  
{ enable :  
true , priority :  
1 , mandatory :  
true , } , backUpShareFactor :  
{ enable :  
true , priority :  
2 , mandatory :  
true , } , socialBackupFactor :  
{ enable :  
true , priority :  
3 , mandatory :  
true , } , passwordFactor :  
{ enable :  
true , priority :  
4 , mandatory :  
true , } , } , } ) Edit this page Previous Custom Authentication Next dApp Share
```