Multi Factor Authentication in PnP Unreal Engine 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.

You can set the mfaLevel within theloginParams 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. Minimum Growth plan required This is a paid feature and the minimum[pricing plan](#) to use this SDK in a production environment is theGrowth Plan . You can use this feature in the development environment for free.

MFALevel

[â](#)

UENUM (BlueprintType) enum

class FMFALevel : uint8 { DEFAULT , OPTIONAL , MANDATORY , NONE } ;

Usage[â](#)

MFASettings

[â](#)

Minimum SCALE plan required This is a paid feature and the minimum[pricing plan](#) to use this SDK in a production environment is theSCALE Plan . You can use this feature in the development environment for free. USTRUCT (BlueprintType) struct

FMfaSettings { GENERATED_BODY ( )

UPROPERTY (EditAnywhere , BlueprintReadWrite) FMfaSetting deviceShareFactor ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FMfaSetting backUpShareFactor ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FMfaSetting socialBackupFactor ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FMfaSetting passwordFactor ;

FMfaSettings ( )

{ } ;

void operator =

(const FMfaSettings & other)

{ deviceShareFactor = other . deviceShareFactor ; backUpShareFactor = other . backUpShareFactor ; socialBackupFactor = other . socialBackupFactor ; passwordFactor = other . passwordFactor ; }

bool operator

(const FMfaSettings & other)

const { if

(deviceShareFactor == other . deviceShareFactor && backUpShareFactor == other . backUpShareFactor && socialBackupFactor == other . socialBackupFactor && passwordFactor == other . passwordFactor) { return true ; } return false ; }

} ;

Usage[â](#)