Using Custom Authentication in PnP Unreal Engine SDK

Custom Authentication is a way to authenticate users with your own custom authentication service. For example, while authenticating with Google, you have the ability to use your own Google Client ID and Dashboard to authenticate users directly. To login using your own custom JWT issuers like Auth0, AWS Cognito, or Firebase, you can add the your configuration to theloginConfig field of theWeb3AuthOptions class.

TheloginConfig field is a key value map. The key should be one of theWeb3AuthProvider in its string form, and the value should be aLoginConfigItem struct instance.

First, configure your own verifier in the Web3Auth Dashboard to use custom authentication.

note This is a paid feature and the minimumpricing plan to use this SDK in a production environment is theGrowth Plan . You can use this feature in the development environment for free. tip Check out how to createCustom Verifier on Web3Auth Dashboard. Then, you should specify the details of your verifier in theLoginConfigItem struct, the details of this struct are as follows.

Argumentsâ

LoginConfigItem

â

- Table
- Interface

Parameter Description verifier The name of the verifier that you have registered on the Web3Auth Dashboard. It's a mandatory field, and acceptsFString as a value. typeOfLogin Type of login of this verifier, this value will affect the login flow that is adapted. For example, if you choosegoogle , a Google sign-in flow will be used. If you choosejwt , you should be providing your own JWT token, no sign-in flow will be presented. It's a mandatory field, and acceptsFTypeOfLogin as a value. clientId Client id provided by your login provider used for custom verifier. e.g. Google's Client ID or Web3Auth's client Id if using 'jwt' as TypeOfLogin. It's a mandatory field, and acceptsFString as a value. name? Display name for the verifier. If null, the default name is used. It acceptsFString as a value. description? Description for the button. If provided, it renders as a full length button. else, icon button. It acceptsFString as a value. verifierSubIdentifier? The field in JWT token which maps to verifier id. Please make sure you selected correct JWT verifier id in the developer dashboard. It acceptsFString as a value. logoHover? Logo to be shown on mouse hover. It acceptsFString as a value. logoLight? Light logo for dark background. It acceptsFString as a value. logoDark? Dark logo for light background. It acceptsFString as a value. mainOption? Show login button on the main list. It acceptsbool as a value. Default value is false. showOnModal? Whether to show the login button on modal or not. Default value is true. showOnDesktop? Whether to show the login button on desktop. Default value is true. showOnMobile? Whether to show the login button on mobile. Default value is true. USTRUCT (BlueprintType) struct

FLoginConfigItem { GENERATED_BODY ( )

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString verifier ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString typeOfLogin ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString name ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString description ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString clientId ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString verifierSubIdentifier ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString logoHover ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString logoLight ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString logoDark ; UPROPERTY (EditAnywhere , BlueprintReadWrite) bool mainOption ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) bool showOnModal ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) bool showOnDesktop ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) bool showOnMobile ;

FLoginConfigItem ( )

{ } ;

bool

operator

(const FLoginConfigItem & other)

{ return other . clientId == clientId ; }

} ;

TypeOfLogin

â

UENUM (BlueprintType) enum

class FTypeOfLogin : uint8 { GOOGLE , FACEBOOK , REDDIT , DISCORD , TWITCH , APPLE , LINE , GITHUB , KAKAO , LINKEDIN , TWITTER , WEIBO , WECHAT , EMAIL_PASSWORDLESS , EMAIL_PASSWORD , JWT } ;

Usageâ

- Google
- Facebook
- Email Passwordless
- Auth0 note
- dApp Share is only returned for the Custom Authentication verifiers.
- Also, 2FA should be enabled for the account using it.
- UsemfaLevel = MFALevel.MANDATORY
- in theLoginParams
- during login. SeeMFA
- for more details. Edit this page Previous Whitelabel Next Multi Factor Authentication