Using PnP Unreal Engine SDK

Once you've installed and successfully initialized Web3Auth, you can use it to authenticate your users.

Logging in a Userâ

web3Auth.login(LoginParams)

â

This function helps your user to trigger the login process. The login flow will be triggered based on the selected provider. You can view the list of selected providers in theprovider section.

Argumentsâ

web3Auth.login() requiresLoginParams as a required input.

LoginParams

â

- Table
- Interface

Parameter Description loginProvider It sets the OAuth login method to be used. You can use any of the supported values areGOOGLE ,FACEBOOK ,REDDIT ,DISCORD ,TWITCH ,APPLE ,LINE ,GITHUB ,KAKAO ,LINKEDIN ,TWITTER ,WEIBO ,WECHAT ,EMAIL_PASSWORDLESS . extraLoginOptions? It can be used to set the OAuth login options for correspondingloginProvider . For instance, you'll need to pass user's email address as. Default value for the field isnull , and it acceptsFExtraLoginOptions as a value. redirectUrl? Url where user will be redirected after successfull login. By default user will be redirected to same page where login will be initiated. Default value for the field isnull , and acceptsFString as a value. appState? It can be used to keep track of the app state when user will be redirected to app after login. Default isnull , and acceptsFString as a value. mfaLevel? Customize the MFA screen shown to the user during OAuth authentication. Default value for field isMFALevel.DEFAULT , which shows MFA screen every 3rd login. It acceptsFMFALevel as a value. dappShare? Custom verifier logins can get a dapp share returned to them post successful login. This is useful if the dapps want to use this share to allow users to login seamlessly. It acceptsFString as a value. curve? It will be used to determine the public key encoded in the jwt token which returned ingetUserInfo function after user login. The default value isFCurve::speck256k1 . USTRUCT (BlueprintType) struct

FLoginParams { GENERATED_BODY ( )

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString loginProvider ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString dappShare ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FExtraLoginOptions extraLoginOptions ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString appState ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString redirectUrl ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FMFALevel mfaLevel ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FCurve curve ;

FLoginParams ( )

{ } ;

FJsonObject getJsonObject ( )

{ FJsonObject output ;

if

(! appState . IsEmpty ( )) output . SetStringField ("appState" , appState) ;

if

(! dappShare . IsEmpty ( )) output . SetStringField ("dappShare" , dappShare) ;

if

(! loginProvider . IsEmpty ( )) output . SetStringField ("loginProvider" , loginProvider) ;

if

(! redirectUrl . IsEmpty ( )) output . SetStringField ("redirectUrl" , redirectUrl) ;

if

(extraLoginOptions . getJsonObject ( )

!=

nullptr) output . SetObjectField ("extraLoginOptions" , extraLoginOptions . getJsonObject ( )) ;

return output ; } } ;

Provider

â

UENUM (BlueprintType) enum

class

FProvider

:

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

ExtraLoginOptions

â

TheLoginParams class accepts theExtraLoginOptions as an optional input, containing advanced options for custom authentication, email passwordless login etc. This parameter can be considered as advanced login options needed for specific cases.

- Table
- Interface

Parameter Description domain? Your custom authentication domain inFString format. For example, if you are using Auth0, it can be example.au.auth0.com. client_id? Client id inFString format, provided by your login provider used for custom verifier. leeway? The value used to account for clock skew in JWT expirations. The value is in the seconds, and ideally should no more than 60 seconds or 120 seconds at max. It takesFString as a value. verifierIdField? The field in JWT token which maps to verifier id. Please make sure you selected correct JWT verifier id in the developer dashboard. It takesFString as a value. isVerifierIdCaseSensitive? Boolean to confirm Whether the verifier id field is case sensitive or not. display? Allows developers the configure the display of UI. It takesFDisplay as a value. prompt? Prompt shown to the user during authentication process. It takesFPrompt as a value. max_age? Max time allowed without reauthentication. If the last time user authenticated is greater than this value, then user must reauthenticate. It takesFString as a value. ui_locales? The space separated list of language tags, ordered by preference. For instancefr-CA fr en . id_token_hint? It denotes the previously issued ID token. It takesFString as a value. id_token? JWT (ID Token) to be passed for login. login_hint? It is used to send the user's email address during Email Passwordless login. It takesFString as a value. acr_values? acc_values scope? The default scope to be used on authentication requests. The defaultScope defined in the Auth0Client is included along with this scope. It takesFString as a value. audience? The audience, presented as the aud claim in the access token, defines the intended consumer of the token. It takesFString as a value. connection? The name of the connection configured for your application. If null, it will redirect to the Auth0 Login Page and show the Login Widget. It takesFString as a value. state? state response_type? Defines which grant to execute for the authorization server. It takesFString as a value. nonce? nonce redirect_uri? It can be used to specify the default url, where your custom jwt verifier can redirect your browser to with the result. If you are using Auth0, it must be whitelisted in the Allowed Callback URLs in your Auth0's application. USTRUCT (BlueprintType) struct

WEB3AUTHSDK_API FExtraLoginOptions { GENERATED_BODY ( )

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString domain ;

UPROPERTY (EditAnywhere , BlueprintReadWrite) FString client_id ;

```
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString leeway ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString verifierIdField ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString max_age ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString ui_locales ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString id_token_hint ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString login_hint ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString acr_values ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString scope ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString audience ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString connection ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString state ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString response_type ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString nonce ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FString redirect_uri ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) bool isVerifierIdCaseSensitive ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FDisplay display ;
UPROPERTY ( EditAnywhere , BlueprintReadWrite ) FPrompt prompt ;
FExtraLoginOptions ( )
{ } ;
TSharedPtr < FJsonObject
getJsonObject ( )
{ TSharedPtr < FJsonObject
```

output

```
MakeShareable ( new FJsonObject ) ;
if
( ! domain . IsEmpty ( ) ) output -> SetStringField ( "domain" , domain ) ;
if
( ! client_id . IsEmpty ( ) ) output -> SetStringField ( "client_id" , client_id ) ;
if
( ! leeway . IsEmpty ( ) ) output -> SetStringField ( "leeway" , leeway ) ;
if
( ! verifierIdField . IsEmpty ( ) ) output -> SetStringField ( "verifierIdField" , verifierIdField ) ;
if
( ! max_age . IsEmpty ( ) ) output -> SetStringField ( "max_age" , max_age ) ;
if
( ! ui_locales . IsEmpty ( ) ) output -> SetStringField ( "ui_locales" , ui_locales ) ;
if
```

```
( ! id_token_hint . IsEmpty ( ) ) output -> SetStringField ( "id_token_hint" , id_token_hint ) ;
if
( ! login_hint . IsEmpty ( ) ) output -> SetStringField ( "login_hint" , login_hint ) ;
if
( ! acr_values . IsEmpty ( ) ) output -> SetStringField ( "acr_values" , acr_values ) ;
if
( ! scope . IsEmpty ( ) ) output -> SetStringField ( "scope" , scope ) ;
if
( ! audience . IsEmpty ( ) ) output -> SetStringField ( "audience" , audience ) ;
if
( ! connection . IsEmpty ( ) ) output -> SetStringField ( "connection" , connection ) ;
if
( ! state . IsEmpty ( ) ) output -> SetStringField ( "state" , state ) ;
if
( ! response_type . IsEmpty ( ) ) output -> SetStringField ( "response_type" , response_type ) ;
if
( ! nonce . IsEmpty ( ) ) output -> SetStringField ( "nonce" , nonce ) ;
if
( ! redirect_uri . IsEmpty ( ) ) output -> SetStringField ( "redirect_uri" , redirect_uri ) ;
if
( output -> Values . IsEmpty ( ) )
{ return
nullptr ; }
output -> SetBoolField ( "isVerifierIdCaseSensitive" , isVerifierIdCaseSensitive ) ;
switch
( display )
{ case FDisplay :: PAGE : output -> SetStringField ( "display" ,
"page" ) ; break ; case FDisplay :: POPUP : output -> SetStringField ( "display" ,
"popup" ) ; break ; case FDisplay :: TOUCH : output -> SetStringField ( "display" ,
"touch" ) ; break ; case FDisplay :: WAP : output -> SetStringField ( "display" ,
"wap" ) ; break ; }
switch
( prompt )
{ case FPrompt :: CONSENT : output -> SetStringField ( "prompt" ,
"consent" ) ; break ; case FPrompt :: LOGIN : output -> SetStringField ( "prompt" ,
"login" ) ; break ; case FPrompt :: NONE : output -> SetStringField ( "prompt" ,
"none" ) ; break ; case FPrompt :: SELECT_ACCOUNT : output -> SetStringField ( "prompt" ,
```

"select_acocunt") ; break ; }

return output ; }

} ;

Curve

â

TheLoginParams class accepts acurve parameter. This parameter can be used to select the elliptic curve to use for the signature.

Sample Responseâ

{ "ed25519PrivKey": "666523652352635....", "privKey": "0ajjsdsd....", "sessionId": "0a652365dsd.....", "userInfo": { "aggregateVerifier": "tkey-google", "email": "john@gmail.com", "name": "John Dash", "profileImage": "https://lh3.googleusercontent.com/a/Ajjjsdsmdjmnm...", "typeOfLogin": "google", "verifier": "torus", "verifierId": "john@gmail.com", "dappShare": "<24 words seed phrase>", // will be sent only incase of custom verifiers "idToken": "", "oAuthIdToken": "", // will be sent only incase of custom verifiers "oAuthAccessToken": "" // will be sent only incase of custom verifiers } }

Usageâ

- Google
- Facebook
- Email Passwordless
- Auth0

Logging out a userâ

web3Auth.logout()

â

Trigger logout flow. This function doesn't take any parameters. A completable future containing void object is returned on successfull logout otherwise an error response is returned.

Usageâ

Working Example Blueprintâ