Custom Authentication in PnP React Native 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 theSdkInitParams object. TheloginConfig field is a key value map. The key should be one of theLOGIN_PROVIDER in its string form, and the value should be a JS Object specified below.

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. Create Custom Verifier Check out how to create aCustom Verifier on the Web3Auth Dashboard. using dapp share * dApp Share is only returned for the Custom verifiers. * Also, 2FA should be enabled for the account using it. UsemfaLevel = MFALevel.MANDATORY * in theLoginParams * during login. SeeMFA * for more details. Then, you should specify the details of your verifier in the value of theLoginConfig map, the details of this map are as follows.

LoginConfig

â

Argumentsâ

- Table
- Interface

Parameter Description verifier The name of the verifier that you have registered on the Web3Auth Dashboard. It's a mandatory field, and acceptsstring 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 acceptsTypeOfLogin 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 acceptsstring as a value. name? Display name for the verifier. If null, the default name is used. It acceptsstring as a value. description? Description for the button. If provided, it renders as a full length button. else, icon button. It acceptsstring 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 acceptsstring as a value. logoHover? Logo to be shown on mouse hover. It acceptsstring as a value. logoLight? Light logo for dark background. It acceptsstring as a value. logoDark? Dark logo for light background. It acceptsstring as a value. mainOption? Show login button on the main list. It acceptsoolean 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. jwtParameters Custom jwt parameters to configure the login. Useful for Auth0 configuration. It acceptsJwtParameters as a value. export

type

LoginConfig

=

Record < string , { verifier :

string ;

/* * *The type of login. Refer to enum*LOGIN_TYPE / typeOfLogin :

TypeOfLogin ;

/* * *Display Name. If not provided, we use the default for openlogin app*/ name ? :

string ;

/* * *Description for button. If provided, it renders as a full length button. else, icon button*/ description ? :

string ;

/* * *Custom client_id. If not provided, we use the default for openlogin app*/ clientId ? :

string ;

verifierSubIdentifier ? :

string ;

/* * Logo to be shown on mouse hover. If not provided, we use the default for openlogin app*/ logoHover ? :

string ;

/* * Logo to be shown on dark background (dark theme). If not provided, we use the default for openlogin app*/ logoLight ? :

string ;

/* * Logo to be shown on light background (light theme). If not provided, we use the default for openlogin app*/ logoDark ? :

string ;

/* * Show login button on the main list*/ mainOption ? :

boolean ;

/* * Whether to show the login button on modal or not*/ showOnModal ? :

boolean ;

/* * Whether to show the login button on desktop*/ showOnDesktop ? :

boolean ;

/* * Whether to show the login button on mobile*/ showOnMobile ? :

boolean ; /* * Custom jwt parameters to configure the login. Useful for Auth0 configuration*/ jwtParameters ? :

JwtParameters ; }

;

typeOfLogin

â

export

type

TypeOfLogin

= |

"google" |

"facebook" |

"reddit" |

"discord" |

"twitch" |

"apple" |

"github" |

"linkedin" |

"twitter" |

"weibo" |

"line" |

"email_password" |

"passwordless" |

"jwt" ;

export

const

LOGIN_PROVIDER

=

{ GOOGLE :

"google" , FACEBOOK :

"facebook" , REDDIT :

"reddit" , DISCORD :

"discord" , TWITCH :

"twitch" , APPLE :

"apple" , LINE :

"line" , GITHUB :

"github" , KAKAO :

"kakao" , LINKEDIN :

"linkedin" , TWITTER :

"twitter" , WEIBO :

"weibo" , WECHAT :

"wechat" , EMAIL_PASSWORDLESS :

"email_passwordless" , SMS_PASSWORDLESS :

"sms_passwordless" , JWT :

"jwt" , }

as

const ;

Exampleâ

- Google
- Facebook
- JWT

import

*

as

WebBrowser

from

"@toruslabs/react-native-web-browser" ; // or import * as WebBrowser from "expo-web-browser"; (for expo)

const web3auth =

new

Web3Auth (WebBrowser ,

{ clientId :

"BC5bANkU4-fil7C5s1uKzRfF0VGqbuaxDQiLnQ8WgF7SEA32lGegAhu7dk4dZf3Rk397blIvfWytXwsRvs9dOaQ" , network :

Network . TESTNET , loginConfig :

{ google :

{ verifier :

"verifier-name" ,

// get it from web3auth dashboard typeOfLogin :

TypeOfLogin . GOOGLE , clientId :

"google_client_id" ,

// google's client id } , } , }) ; import

*

as

WebBrowser

from

"@toruslabs/react-native-web-browser" ; // or import * as WebBrowser from "expo-web-browser"; (for expo)

const web3auth =

new

Web3Auth (WebBrowser ,

{ clientId :

"BC5bANkU4-fil7C5s1uKzRfF0VGqbuaxDQiLnQ8WgF7SEA32lGegAhu7dk4dZf3Rk397blIvfWytXwsRvs9dOaQ" , network :

Network . TESTNET , loginConfig :

{ facebook :

{ verifier :

"verifier-name" ,

// get it from web3auth dashboard typeOfLogin :

TypeOfLogin . FACEBOOK , clientId :

"facebook_client_id" ,

// facebook's client id } , } , }) ; import

*

as

WebBrowser

from

"@toruslabs/react-native-web-browser" ; // or import * as WebBrowser from "expo-web-browser"; (for expo)

const web3auth =

new

Web3Auth (WebBrowser ,

{ clientId :

"BC5bANkU4-fil7C5s1uKzRfF0VGqbuaxDQiLnQ8WgF7SEA32lGegAhu7dk4dZf3Rk397blIvfWytXwsRvs9dOaQ" , network :

Network . TESTNET , loginConfig :

{ jwt :

{ verifier :

"verifier-name" ,

// get it from web3auth dashboard typeOfLogin :

TypeOfLogin . JWT , clientId :

"web3auth's client id" ,

// web3auth's client id } , } , }) ;

ExtraLoginOptions

for special login methodsâ

In addition to theLoginConfig you can pass extra options to thelogin function to configure the login flow for cases requiring additional info for enabling login. TheExtraLoginOptions accepts the following parameters:

- Table
- Interface

Parameter Description additionalParams? Additional params in[key: string] format for OAuth login, use id_token(JWT) to authenticate with web3auth. domain? Your custom authentication domain instring format. For example, if you are using Auth0, it can be example.au.auth0.com. client_id? Client id instring 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 takesstring 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 takesstring 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 takesstring as a value. prompt? Prompt shown to the user during authentication process. It takesstring 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 takesstring 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 takesstring 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 takesstring 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 takesstring as a value. audience? The audience, presented as the aud claim in the access token, defines the intended consumer of the token. It takesstring 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 takesstring as a value. 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. * ExtraLoginOptions * BaseLoginOptions

export

interface

ExtraLoginOptions

extends

BaseLoginOptions

{ /* * *Your Auth0 account domain such as'example.auth0.com', * 'example.eu.auth0.com' or , 'example.mycompany.com' * (when using custom domains)* / domain ? :

string ; /* * *The Client ID found on your Application settings page*/ client_id ? :

string ; /* * *The default URL where Auth0 will redirect your browser to with * the authentication result. It must be whitelisted in * the "Allowed Callback URLs" field in your Auth0 Application's * settings. If not provided here, it should be provided in the other * methods that provide authentication.* / redirect_uri ? :

string ; /* * *The value in seconds used to account for clock skew in JWT expirations. * Typically, this value is no more than a minute or two at maximum. * Defaults to 60s.* / leeway ? :

number ; /* * *The field in jwt token which maps to verifier id*/ verifierIdField ? :

string ; /* * *Whether the verifier id field is case sensitive * @defaultValue true*/ isVerifierIdCaseSensitive ? :

boolean ; } export

interface

BaseLoginOptions

{ /* * *If you need to send custom parameters to the Authorization Server, * make sure to use the original parameter name.*/ [key :

string] :

unknown ; /* * *- 'page': displays the UI with a full page view * -'popup': displays the UI with a popup window * -'touch': displays the UI in a way that leverages a touch interface * - 'wap': displays the UI with a "feature phone" type interface*/ display ? :

"page"

|

"popup"

|

"touch"

|

"wap"

|

string ; /* * *- 'none': do not prompt user for login or consent on re-authentication * -'login': prompt user for re-authentication * - 'consent': prompt user for consent before processing request * -'select_account': prompt user to select an account*/ prompt ? :

"none"

|

"login"

|

"consent"

|

"select_account"

|

string ; /* * *Maximum allowable elapsed time (in seconds) since authentication. * If the last time the user authenticated is greater than this value, * the user must be re-authenticated.* / max_age ? :

string

|

number ; /* * *The space-separated list of language tags, ordered by preference. * For example:'fr-CA fr en'.* / ui_locales ? :

string ; /* * *Previously issued ID Token.*/ id_token_hint ? :

string ; /* * *The user's email address or other identifier. When your app knows * which user is trying to authenticate, you can provide this parameter * to pre-fill the email box or select the right session for sign-in. * * This currently only affects the classic Lock experience.* / login_hint ? :

string ; acr_values ? :

string ; /* * *The default scope to be used on authentication requests. * The defaultScope defined in the Auth0Client is included * along with this scope* / scope ? :

string ; /* * *The default audience to be used for requesting API access.*/ audience ? :

string ; /* * *The name of the connection configured for your application. * If null, it will redirect to the Auth0 Login Page and show * the Login Widget.* / connection ? :

string ; }

Using Auth0 Loginâ

Auth0 has a special login flow, called the SPA flow. This flow requires aclient_id anddomain to be passed, and Web3Auth will get the JWTid_token from Auth0 directly. You can pass these configurations in theExtraLoginOptions object in the login function.

const web3auth =

new

Web3Auth (WebBrowser ,

{ clientId :

"YOUR_CLIENT_ID" ,

// web3auth's client id network :

OPENLOGIN_NETWORK . TESTNET ,

// or other networks loginConfig :

{ jwt :

{ verifier :

"verifier-name" ,

// get it from web3auth dashboard for auth0 configuration typeOfLogin :

"jwt" , clientId :

"your auth0 client id" ,

// get it from auth0 dashboard } , } , }) ; web3auth . login ({ loginProvider :

LOGIN_PROVIDER . JWT , redirectUrl : resolvedRedirectUrl ,

// redirect url after login extraLoginOptions :

{ domain :

"example.com" ,

// domain of your auth0 app verifierIdField :

"sub" ,

// The field in jwt token which maps to verifier id. } , }) ;

Custom JWT Loginâ

If you're using any other provider like Firebase/ AWS Cognito or deploying your own Custom JWT server, you need to put the jwt token into theid_token field of theextraLoginOptions .

const web3auth =

new

Web3Auth (WebBrowser ,

{ clientId :

"YOUR_CLIENT_ID" ,

// web3auth's client id network :

Network . TESTNET , loginConfig :

{ jwt :

{ verifier :

"verifier-name" ,

// get it from web3auth dashboard typeOfLogin :

TypeOfLogin . JWT , clientId :

"YOUR_CLIENT_ID" ,

// web3auth's client id } , } , }) ; web3auth . login ({ loginProvider :

LoginProvider . JWT , redirectUrl : resolvedRedirectUrl ,

// redirect url after login extraLoginOptions :

{ id_token :

"your JWT id token" , verifierIdField :

"sub" ,

// auth0 generally uses sub as unique identifier } , }) ;

Email Passwordlessâ

To use theEMAIL_PASSWORDLESS login, you need to put the email into thelogin_hint field of theextraLoginOptions .

web3auth . login ({ loginProvider :

LoginProvider . EMAIL_PASSWORDLESS , redirectUrl : resolvedRedirectUrl , extraLoginOptions :

{ login_hint :

"hello@web3auth.io" , } , }) ;Edit this page Previous Whitelabel Next Multi Factor Authentication