Installing PnP React Native SDK

Selecting your Workflowâ

In React Native, you have the choice to use one of the following workflows:

Expo Managed Workflowâ

Your Expo app is built on your Expo's cloud, so you don't have control over the native modules used in the app. Developers build managed workflow apps usingexpo-cli on their computers and a development client on their mobile devices. Managed workflow apps typically use one or more Expo services, such as push notifications, builds, and updates.

warning Web3Auth SDKs are not compatible with "Expo Go" app. They are compatible only with Custom Dev Client and EAS builds. Please refer to the troubleshooting section for more on this.

Please runnpx expo prebuild to generate native code based on the version of expo a project has installed, before moving forward.

Bare React Native Workflowâ

Your Bare React Native app is entirely built on your machine. In this workflow, the developer has complete control, along with the complexity that comes with that. Configuration withapp.json /app.config.js is mostly not supported in this context; instead, you will need to configure each native project directly with Swift/Kotlin native modules. Check out thetroubleshooting section for fixing common issues.

tip You can read more about different workflows in theExpo documentation .

Installationâ

@web3auth/react-native-sdk

â

- npm
- Yarn
- pnpm

npm install --save @web3auth/react-native-sdk yarn add @web3auth/react-native-sdk pnpm add @web3auth/react-native-sdk

Adding a Web Browser Moduleâ

We will also require aWebBrowser implementation to allow our JS-based SDK to interact with the native APIs, and there are different extra installation steps depending on whether you are using the bare workflow or managed workflow.

- Expo Managed Workflow
- Bare React Native Workflow

Expo Web Browserâ

When using our SDK with an Expo-based React Native app (aka managed workflow), you have to install theexpo-web-browser package as aWebBrowser implementation.

expo install expo-web-browser

React Native Web Browserâ

When using our SDK with a bare workflow React Native app, you have to install aWebBrowser implementation made by us.

- npm
- Yarn
- pnpm

npm install --save @toruslabs/react-native-web-browser yarn add @toruslabs/react-native-web-browser pnpm add @toruslabs/react-native-web-browser

Adding aStorage

Moduleâ

Now with v4, we need to pass aStorage parameter to the SDK, which will be used for session management without storing the private keys of the user in the device.

- Expo Managed Workflow
- Bare React Native Workflow

Expo Secure Storeâ

When using our SDK with an Expo-based React Native app (aka managed workflow), you have to install theexpo-secure-store package as aStorage implementation.

expo install expo-secure-store

React Native Encrypted Storageâ

When using our SDK with a bare workflow React Native app, you have to install aStorage implementation provided by react-native.

- npm
- Yarn
- pnpm

npm install --save react-native-encrypted-storage yarn add react-native-encrypted-storage pnpm add react-native-encrypted-storage

Configurationâ

After you have installed the files needed for your workflow, you'll have to configure the SDK with some additional steps to be able to use the SDK properly.

Expo Managed Workflowâ

- Adding URL scheme toapp.json

To allow the Expo-based SDK to work with exported Expo Android apps, you need to add the designated scheme intoapp.json

app.json { "expo": { "scheme": "web3authexpoexample" // replace with your own scheme } } * For constructing yourredirectUrl * , you'll need to use theexpo-linking * , which will help you generate aredirectUrl * for your app. Make sure you * register that URL in theWeb3Auth Developer Dashboard * .

App.js import

Constants ,

{

AppOwnership

}

from

"expo-constants" ; import

*

as

Linking

from

"expo-linking" ;

const resolvedRedirectUrl = Constants . appOwnership

==

AppOwnership . Expo

||

Constants . appOwnership

==

AppOwnership . Guest ?

Linking . createURL ("web3auth" ,

{ }) :

Linking . createURL ("web3auth" ,

{ scheme : scheme }) ; tip You may refer tothese example apps and try it out yourself.

Bare React Native Workflowâ

For the bare workflow, you need to perform additional installation steps, alongside specific configurations for Android and iOS separately.

Androidâ

- Make sure that the minimum SDK compile version inbuild.gradle
- is 31 or more.

android/build.gradle buildToolsVersion = "31.0.0" minSdkVersion = 21 compileSdkVersion = 31 targetSdkVersion = 31 * Add the intent filter withscheme * defined in yourAndroidManifest.xml

android/app/src/main/AndroidManifest.xml < intent-filter

< action

android: name = " android.intent.action.VIEW "

/> < category

android: name = " android.intent.category.DEFAULT "

/> < category

android: name = " android.intent.category.BROWSABLE "

/>

replace with your own scheme

< data

android: scheme = " web3authrnexample "

/> </ intent-filter

- SDK version 31 requires you to explicitly defineandroid:exported="true"
- inAndroidManifest.xml
- , and check whether it is correctly present or
- not.

android/app/src/main/AndroidManifest.xml < activity android: name = " .MainActivity " android: label = " @string/app_name " android: configChanges = " keyboard|keyboardHidden|orientation|screenSize|uiMode " android: launchMode = " singleTask " android: windowSoftInputMode = " adjustResize " android: exported = " true "

- YourredirectUrl
- is your defined scheme following some identifier or your choice. For example, if your scheme isweb3authrnexample
- , you can
- define yourredirectUrl
- asweb3authrnexample://openlogin

- . Make sure you register thisredirectUrl
- in theWeb3Auth Developer Dashboard
- .

App.js const scheme =

"web3authrnexample" ;

// Or your desired app redirection scheme const resolvedRedirectUrl =

{ scheme } ://openlogin ;

iOSâ

- Make sure that the minimum SDK compile version inPodfile
- is 14 or more.

ios/Podfile platform :ios, '14' * Install the Cocoa Pods within your project directory.

cd ios pod install * In iOS, you don't need to addredirectUrl * as an iOS Custom URL Scheme, however, you may add it to yourInfo.plist * , but it is not required. Make * sure yourredirectUrl * is registered in theWeb3Auth Developer Dashboard * .

tip You may refer tothese example apps and try it out yourself.

Troubleshootingâ

Bundler Issues: Missing Dependenciesâ

You might face issues mentioning that certain dependencies are missing within the React Native environment. These are node dependencies that need to be polyfilled in your application, to enable Web3Auth functionalities. Furthermore, your bundler needs to be reconfigured to use them while building the app. Please check out ourReact Native Troubleshooting Guide Edit this page Previous Web3Auth PnP React Native SDKNext Initialize