

# Installing Core Kit SFA React Native SDK

## Selecting your Workflow<sup>â</sup>

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

### Expo Managed Workflow<sup>â</sup>

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 using `expo-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 run `expo prebuild` to generate native code based on the version of expo a project has installed, before moving forward.

### Bare React Native Workflow<sup>â</sup>

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 with `app.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 the [troubleshooting section](#) for fixing common issues.

tip You can read more about different workflows in the [Expo documentation](#).

## Installation<sup>â</sup>

[@web3auth/single-factor-auth-react-native](#)

<sup>â</sup>

- npm
- Yarn
- pnpm

```
npm install --save @web3auth/single-factor-auth-react-native yarn add @web3auth/single-factor-auth-react-native pnpm add @web3auth/single-factor-auth-react-native
```

## Adding aStorage

Module<sup>â</sup>

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 Store<sup>â</sup>

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

```
expo install expo-secure-store
```

### React Native Encrypted Storage<sup>â</sup>

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

## Configuration

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 Workflow

- Adding URL scheme to app.json

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

app.json { "expo": { "scheme": "web3authexpoexample" // replace with your own scheme } } tip You may refer [these example apps](#) and try it out yourself.

### Bare React Native Workflow

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

#### Android

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

android/build.gradle buildToolsVersion = "31.0.0" minSdkVersion = 21 compileSdkVersion = 31 targetSdkVersion = 31 \* Add the intent filter with scheme \* defined in your AndroidManifest.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 define android:exported="true"
- in AndroidManifest.xml
- , 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 "
```

#### iOS

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

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

cd ios pod install tip You may refer to [these example apps](#) and try it out yourself.

## Troubleshooting

### Bundler Issues: Missing Dependencies

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 our [React Native Troubleshooting Guide](#) [Edit this page](#) [Previous](#) [Web3Auth Core Kit Single Factor Auth React Native SDK](#) [Next](#) [Initialize](#)