EcomLabs

STEP-BY-STEP GUIDE

Expo Router v56: migrating off React Navigation

Moving an Expo Router app from SDK 55 to SDK 56, in order: an import inventory, the SDK upgrade, the codemod, what you fix by hand, third-party libraries and testing on iOS and Android. Every command, plus the checklist.

FOR	Expo apps using Expo Router on SDK 55 that are moving to SDK 56
TESTED WITH	Node 22 · expo-codemod 58.0.1 on test files · tsc --strict against the expo-router 56.2.22 types
UPDATED	November 19, 2026
DOCS	docs.expo.dev/router/migrate/sdk-55-to-56
ARTICLE	ecomlabs.dev/g/router-en
AUTHOR	Luis Morales · @luismoralesdev

Contents

Before you start	What changes
Step 01	Inventory your imports
Step 02	Upgrade the SDK
Step 03	Run the codemod
Step 04	What you fix by hand
Step 05	Third-party libraries
Step 06	Test on iOS and Android
Next	SDK 57 and SDK 58
When you finish	Checklist

BEFORE YOU START

What changes

Since SDK 56, Expo Router no longer depends on React Navigation: Expo forked the parts it used and moved them into the package. According to the official guide:

- Your app can no longer import from `@react-navigation/*`; if it does, the Expo CLI 56 bundler stops with an error.
- The API does not change: only the import paths move.
- Libraries in `node_modules` that import `@react-navigation/core` are rewritten automatically, as a temporary measure.

Work on a new branch with a clean tree: the codemod rewrites files and you will read the whole diff.

STEP 01

Inventory your imports

The first command lists every React Navigation import in your code; the second, which dependencies bring React Navigation in (with yarn or pnpm, `yarn why` or `pnpm why`):

```
grep -rnE "from ['\"]@react-navigation/" app src components
npm ls @react-navigation/core @react-navigation/native
```

Write down what the codemod does not handle: imports from `native-stack` or `drawer`, namespace or default imports, and `createStackNavigator` or `TransitionPresets` from `@react-navigation/stack`.

STEP 02

Upgrade the SDK

Expo recommends upgrading one SDK at a time. From 55:

```
npm install expo@^56.0.0
npx expo install --fix
npx expo-doctor
```

The first installs SDK 56; the second brings the other packages, `expo-router` included (now 56.x, as the version follows the SDK), to that SDK's versions; the third checks for common problems and, on SDK 56, warns if `react-navigation` is still installed next to `expo-router`. You clear that warning in step 05.

STEP 03

Run the codemod

From the project root. Pass every folder with code, or a quoted glob. With other package managers, replace `npx` with `yarn dlx`, `pnpm dlx` or `bunx`:

```
npx expo-codemod sdk-56-expo-router-react-navigation-replace app components
```

At the end it prints `ok`, `skipped` and `errors`. What we saw with `expo-codemod` 58.0.1:

- **ok:** file rewritten. Imports with the same target are merged into one and keep their `type` markers.
- **skipped:** the file had `import * as`, a default import, or an import from `native-stack` or `drawer`. None of its imports were touched.

> **CAUTION**
> The codemod writes the paths with double quotes, and on Windows it left CRLF line endings. Run your formatter and read `git diff`.

STEP 04

What you fix by hand

BEFORE	AFTER
`@react-navigation/native`, `core`, `elements`, `routers`	`expo-router/react-navigation`
`@react-navigation/stack`	`expo-router/js-stack`
`@react-navigation/bottom-tabs`	`expo-router/js-tabs`
`@react-navigation/material-top-tabs`	`expo-router/js-top-tabs`
`@react-navigation/native-stack`	The `Stack` layout from `expo-router`
`@react-navigation/drawer`	The `Drawer` layout from `expo-router/drawer`

Namespace import. Change `import * as Nav from '@react-navigation/native'` to named imports (`import { useNavigation } from ...`) and run the codemod again on that file.

native-stack. In Expo Router the stack is a file-based layout: a `_layout.tsx` in the screens' folder replaces `createNativeStackNavigator`, and every file in the folder is a screen. For `drawer`, use `Drawer` from `expo-router/drawer`.

```
// app/_layout.tsx
import { Stack } from 'expo-router';

export default function Layout() {
  return <Stack />;
}
```

> **CAUTION**
>
> On SDK 56, `expo-router/js-stack` only exports `Stack` (we checked the 56.0.0 and 56.2.22 types). `createStackNavigator` and `TransitionPresets` are not there and the import fails in TypeScript.

Run the step 01 `grep` again. If it prints nothing, your code is migrated.

STEP 05

Third-party libraries

If a library imports `@react-navigation/core`, Expo CLI 56 and 57 rewrite the import to `expo-router/react-navigation`. If it imports `@react-navigation/native` or another package, they do not: look for a newer release, test the screen where you use it and, if it fails, open an issue with the library. To isolate a problem:

```
EXPO_ROUTER_DISABLE_RN_NAVIGATION_CHECK=1 npx expo start
```

That variable turns off the rewrite and also the error for stale imports in your code. Do not leave it in `.env` or CI. When you are done, remove the `@react-navigation/*` packages from your `package.json` and check again:

```
npm uninstall @react-navigation/native @react-navigation/elements
npm ls @react-navigation/core @react-navigation/native
npx expo-doctor
```

List the ones you have in `npm uninstall`. If `npm ls` still shows one, a library brings it in.

STEP 06

Test on iOS and Android

```
npx expo start --clear
npx expo run:ios
npx expo run:android
```

`--clear` starts Metro with an empty cache; `run:ios` builds locally and needs a Mac with Xcode; `run:android` builds the Android app. On both platforms check light and dark theme, headers and the back button, tabs, deep links, screens with `useIsFocused` or `useFocusEffect`, and every screen that uses a library from step 05.

NEXT

SDK 57 and SDK 58

SDK 57 (June 30, 2026) has no breaking changes in Expo Router. SDK 58 (October 15, 2026) rebuilt the navigation core and removed most of the API copied from React Navigation; its guide is at docs.expo.dev/router/migrate/sdk-57-to-58.

> **CAUTION**
>
> Expo CLI 58.0.2 removed the `@react-navigation` checks: on SDK 58 there is no error for stale imports and no rewrite for libraries. Keep the step 01 `grep` in your CI.

WHEN YOU FINISH

Checklist

- [ ] New branch, import and library inventory done.
- [ ] `expo` on `^56.0.0` and `npx expo install --fix` run.
- [ ] Codemod run on every folder; skipped files fixed by hand.
- [ ] `native-stack` and `drawer` moved to the `Stack` and `Drawer` layouts.
- [ ] The `grep` for old imports prints nothing.
- [ ] `npx expo-doctor` is clean and libraries are tested.
- [ ] The app builds and works on iOS and Android.
- [ ] `EXPO_ROUTER_DISABLE_RN_NAVIGATION_CHECK` is not left set.