

How to use Expo with React Native

BY ROBIN WIERUCH

- [EDIT THIS POST](#)

-  [Follow on Twitter](#) 16k
- [Follow on Facebook](#)

For all my React Native tutorials, I intend to use Expo as baseline, because it just makes things so much easier when developing an app with React Native. If you want to follow my tutorials about React Native, you should get this setup right.

REACT NATIVE WITH EXPO

Expo is a powerful environment for React Native which helps you from creation to distribution of your React Native apps. Without the need to directly use XCode (iOS) or Android Studio, Expo allows you to see your work in progress apps on a phone, emulator, or browser. It's fair to say that Expo makes React Native development much easier, because it takes all the friction away. When you write code in Expo, you still write React Native code, but with the support of the Expo CLI and Expo Client on your phone. We will explore both in a bit.

For the installation of Expo, your best bet may be following the [installation instructions](#) from the official Expo documentation, because its dependencies change from time to time. However, I want to try to keep these in sync here, so you may want to follow my instructions as well.

First, you have to install [Node.js](#). This makes [npm](#) -- the Node.js package manager -- and node itself available on the command line. Afterward, install the Expo CLI (command line interface) globally on your machine:

```
npm install expo-cli --global
```

You are ready to create your first project. Use the Expo command to initialize a new project on the command line. The name you are giving your project is up to you:

```
expo init myReactNativeProject
```

Afterward, navigate into your project and start it:

```
cd myReactNativeProject  
expo start
```

After your project started, you have multiple options to develop your React Native app. The most popular one may be starting the app directly on your iOS or Android device by simply scanning the QR

code after the startup with your phone's camera app. You will have to install the Expo Client from your App Store on your phone first. Other options include starting the app in your browser or on a phone emulator. Regardless of which way you choose, after starting the app make sure that it's up and running on your desired device.

EXPO CONFIGURATION

Expo is a powerful suite for React Native apps, because it hides plenty of the more complex tooling for you. There are several things you can do to optimize your Expo coding experience though.

First, I like to move the entry point file -- which is by default *App.js* -- to another location. Therefore, create a new *index.js* file in your project's root folder and give it the following implementation details:

```
import { registerRootComponent } from 'expo';

import App from './src/App';

registerRootComponent(App);
```

In order to instruct Expo to take this file instead of the default *App.js* file in your project's root folder, open your *package.json* file and change the *main* property:

```
{
  "main": "index.js",
  "scripts": {
    ...
  },
  "dependencies": {
    ...
  },
  ...
}
```

Last but not least, create a new *src/* folder with a *src/App.js* file. Move your code from the old *App.js* file to your new *src/App.js* file and delete the old *App.js* file. Now, everything should work as before, but you have the benefit of having a dedicated source code folder for your React Native app.

If you have any other Expo configurations that you want to share with others, please write a comment below. After all, I hope you are ready to start coding with your React Native application by using Expo.