Techiediaries

HANDS-ON REACT NATIVE

LEARN REACT NATIVE STEP BY STEP

By Ahmed Bouchefra

React Native for Beginners

Learn to build a mobile application from scratch using React Native

Ahmed Bouchefra

This book is for sale at http://leanpub.com/react-native-for-beginners

This version was published on 2021-01-09

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Chapter One

In this book, we'll teach you to build your first React Native mobile app from scratch going through the essential concepts.

We'll be using React Native v0.6, the latest version as of this writing.

⚠

You will learn how to:

- Create a React Native project,
- Create views with React components and Flexbox layout,
- Add navigation,
- Style UIs,
- Fetch data from remote servers and APIs,
- Get user input,
- Listen to touch events,
- Add images, etc.

React Native vs. Cordova and Ionic

Unlike Cordova, Ionic and other hybrid mobile solutions, you can use React Native for writing truly-native mobile applications for Android and iOS using JavaScript, React, JSX and a bridge that invokes the native rendering APIs for the target mobile system.

As a result, your application is actually a native application that looks and feels like the native host UI and not a web application that runs inside a web-view. This means, your app will have increased performance that can be slightly equaled to the performance of apps built using Java for Android and Swift for iOS.

> **Note**: React Native apps fall slightly in terms of performance behind apps built with native languages like Java and Swift because we still have the JS bridge between the app and the native APIs.

Prerequisites

You will need the following prerequisites to successfully follow this tutorial.

Modern ES6 features

React Native is built on top of React - A popular JavaScript front-end library for building user interfaces, so you will need to know JavaScript including the latest ES6 features such as imports, exports and arrow functions.

React basics

React Native is simply **React** + a **Native Bridge**.

So if you learn React, you'll also end up learning most of React Native and vice versa!

Except that for React, you'll need to use HTML and for React Native, you need to use some predefined UI components designed for mobile UIs.

You need to know about:

- Props and state,
- The component life-cycle methods,
- How to create React (class-based or function-based) components,
- React Hooks,
- JSX.

JSX allows you to write XML markup inside JavaScript files. According to the official website of React[1], it's defined as:

> A syntax extension to JavaScript. We recommend using it with React to describe what the UI should look like. JSX may remind you of a template language, but it comes with the full power of JavaScript.

React hooks — `useState`, `useEffect` and `useContext`

React is now leaning toward a function-based approach for writing components with the introduction of React Hooks which allow you to access the state and other features, which were only available in class-based components before, in your function components.

Props, state and context

Props, state and context are important concepts in React.

React Native components have predefined props that allow users to configure them. For example, all components have the `style` prop that's used to apply styles.

[1] https://reactjs.org/docs/introducing-jsx.html

Node.js and NPM

The React Native CLI is based on Node.js, so you will need to have Node.js 8.3+ and NPM installed on your machine.

See how you can install Node.js using a package manager[2].

Testing on Android and Ubuntu 19.04

We'll be testing our app inside an Android emulator installed on a Ubuntu 19.04 system. So you will need to have:

- JDK 8+ and Android Studio installed on your machine.
- The `ANDROID_HOME` environment variable set to the path of your Android SDK.

On Ubuntu, you can add Android SDK to your `PATH` by adding the following lines to the `~/.bash_profile` or `~/.bashrc` configuration files:

```
export ANDROID_HOME=$HOME/Android/Sdk
export PATH=$PATH:$ANDROID_HOME/emulator
export PATH=$PATH:$ANDROID_HOME/tools
export PATH=$PATH:$ANDROID_HOME/tools/bin
export PATH=$PATH:$ANDROID_HOME/platform-tools
```

We also added the `tools/bin` and `platform-tools` folder to the `PATH` variable to be able to run various utilities like `adb` from any folder. Next, run the following command:

```
$ source ~/.bashrc
```

> Android Debug Bridge (adb) is a versatile command-line tool that lets you communicate with a device. The adb command facilitates a variety of device actions, such as installing and debugging apps, and it provides access to a Unix shell that you can use to run a variety of commands on a device. Source[3]

As the time of this writing, the latest version of Android SDK Platform Tools version is v29 which has a bug that causes the **adb: error: cannot bind listener: Operation not permitted** and prevents the hot-reload of the changes in the emulator when updating the source code. As of now, you can solve this problem by downgrading your tools to v28 as follows:

- Open Android Studio and go to `SDK Tools`,

[2] https://nodejs.org/en/download/package-manager/
[3] https://developer.android.com/studio/command-line/adb

- Uncheck `Android SDK Platform Tools` and click `Apply`
- Download `Android SDK Platform Tools 28` using this link[4].
- Unzip the `platform-tools` folder into your Android SDK folder.

See the adb: error: cannot bind listener: Operation not permitted[5] and Cannot bind listener: operation not permitted[6] issues for more information.

Starting an Android Emulator

First, let's see the installed AVDs in our system:

```
$ emulator -list-avds
```

In my case, I get the following output:

```
Pixel_2_XL_API_28
```

> **Note**: AVDs stands for Android Virtual Devices. They are not available by default after you install Android Studio. So you'll need to create an AVD before you can use it to start an emulator.

Next, you can start your emulator with a specified AVD, using the following command:

```
$ emulator -avd Pixel_2_XL_API_28
```

For more information, you can also refer to Start the emulator from the command line[7].

Visual Studio Code

We'll be using Visual Studio Code as our source code editor. It's available for Windows, Linux and MAC. It has built-in support for JavaScript, TypeScript and Node.js. You can simply go to the official website[8] and download the right binaries for your system.

On Ubuntu 19.04, you can also install it from the Snap store by running the following command:

```
$ sudo snap install code
```

[4] https://dl.google.com/android/repository/platform-tools_r28.0.0-linux.zip

[5] https://github.com/react-native-community/cli/issues/437

[6] https://github.com/microsoft/vscode-react-native/issues/1028

[7] https://developer.android.com/studio/run/emulator-commandline

[8] https://code.visualstudio.com/

For iOS development

To target iOS, you will need to have a macOS system, next you need to get ****an iOS developer's account. It's available for free, but only for development. ****

> **Note**: To deploy the app to the iOS App Store, you will need to get a license with $99/year. ****

Next, you need to download ****and install Xcode, from the App Store or from the official Xcode website[9].

> **Note**: Xcode includes the Xcode IDE, the iOS simulators, and the iOS SDK.

Creating your first React Native project

Open a new terminal and run the following code to invoke the React Native CLI[10] using the `npx` command:

```
$ npx react-native init firstapp
```

> **Note**: Before continuing, you should, by now, have started your Android emulator or connected a real device to your machine.

Next, start the Metro Bundler using the following commands:

```
$ cd firstapp
$ react-native start
```

> **Note**: Metro[11] is a JavaScript bundler for React Native which is fast, scalable and integrated that compiles your React Native code (ES6+) to JavaScript (ES5) using Babel.

You need to leave Metro running. So open a new terminal and run the following commands to compile and launch your application in the Android emulator:

```
$ cd firstapp
$ react-native run-android
```

The `run-android` command will compile and install the app in Android.

[9] https://developer.apple.com/xcode/
[10] https://github.com/react-native-community/cli
[11] https://facebook.github.io/metro/

> **Note**: If you are under a macOS and you want to install the app in an iOS emulator or real device, you need to use the `react-native run-ios` command instead.

You need to wait for your app to be built. You'll finally get the **BUILD SUCCESSFUL** message in your terminal and your app should be opened in the emulator or a real device if attached.

This is a screenshot:

React Native App in the Android Emulator

This is a screenshot of our app running inside an Android emulator:

React Native App on Android

As, we previously mentioned, if you have Android SDK Platform Tools v29, you may get **adb**: **error**: **cannot bind listener**: **Operation not permitted**. In this case, you need to downgrade to v28. Refer the previous **Testing on Android and Ubuntu 19.04** section.

Opening your project in Visual Studio Code

After running your app inside the Android emulator. Run the following command to open the project in Visual Studio Code from your project's folder:

```
$ code .
```

React Native Project in VS Code

The anatomy of a React Native project

In the left panel, we can see the structure of our project. It includes the typical folders and configuration files for a Node.js project such as the `package.json` and `package-lock.json` files and the `node_modules` folder. We have also other configuration files such as:

- `babel.config.js`: The configuration file for Babel (A compiler and transpiler for JavaScript)
- `metro.config.js`: The configuration file for Metro, a JavaScript bundler for React Native,

- `app.json`: configures parts of our app that don't belong in code. See this article[12].
- `watchman.config`: The configuration file for Watchman[13], a file watch service,
- `.flowconfig`: The configuration file for Flow[14], a static type checker for JavaScript,
- `.eslintrc.js`: The configuration file for ESLint[15], a JavaScript and JSX linter (a tool for code quality),
- `.buckconfig`: The configuration file for Buck[16], a build system created by Facebook,
- `.gitignore` and `.gitattributes`: ignores all files in version control that should be unique to each development machine

We have the following folders:

- `android`: The folder for the Android project,
- `ios`: The folder for the iOS project,
- `__tests__`: The folder for tests.

We have the following JavaScript files:

- `App.js`: The main component in our React Native app,
- `index.js`: The main file of our application where the components are registered.

[12] https://docs.expo.io/versions/latest/workflow/configuration/
[13] https://facebook.github.io/watchman/
[14] https://flow.org/
[15] https://eslint.org/
[16] https://buck.build/

Writing your first component — View & Text

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

View and Text

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

How to style components in React Native

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Defining styles with `StyleSheet`

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Creating layouts with Flexbox

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Flex

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Justify Content

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Align Items

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Adding a background image with `<Image>` & `<ImageBackground>`

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Networking and Conditional Rendering

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Conditional Rendering & Multiple Views

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Networking using Fetch in React Native

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Displaying lists

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Displaying lists of data in React Native using `FlatList`

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Buttons

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Creating buttons with the Button & Touchable components and listening for touch events

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Styling buttons

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Adding item separators and header to FlatList

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Storage and Linking

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Saving data in local databases with `AsyncStorage` & React Native

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Opening external links with Linking and React Native

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Navigation

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Adding Navigation using React Navigation

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

React Navigation using SwitchNavigator

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.

Adding stack navigation with StackNavigator

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/react-native-for-beginners.