React Native - Install & Config - OS X

- Dr Nick Hayward

Install and configure React Native on OS X for Android and iOS development.

Contents

Intro

React Native on OS X includes support for native development with Android and iOS.

Further details on React Native can be found at the following URL,

- https://facebook.github.io/react-native/docs/getting-started.html

and then select the link for *Building Projects with Native Code*.

For OS X options, simply select the button for *macOS*, and either *iOS* or *Android* to see further details.

Basic dependencies

React Native itself needs Node.js installed,

- https://nodejs.org/en/

and a tool developed by Facebook called *Watchman*. This tool monitors changes in the local filesystem,

- https://facebook.github.io/watchman/

Each of these tools may be installed on OS X using the *Homebrew* package manager,

- https://brew.sh/

Brew install commands are as follows,

```
brew install node
brew install watchman
```

n.b. Node.js has to be at least version 4 or higher.

React Native CLI

We can then install the command line tool for React Native.

Issue the following terminal command to install React Native CLI,

```
npm install -g react-native-cli
```

To check it's installed correctly, issue the following terminal command

```
react-native -v
```

Android install

For Android development, we need to install a recent version of the JDK (Java SE Development Kit), which is available at the following URL,

- http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Then, we need to download, install, and configure a local development environment for Android. This can become a tad fiddly, and needs to be completed as follows.

1. download and install the latest version of Android Studio from the following URL,

- https://developer.android.com/studio/index.html

For the install itself, ensure the following options have been selected for a *custom* setup,

- Android SDK
- Android SDK Platform
- Performance (Intel HAXM)
- Android Virtual Device (AVD)

Then, go ahead and install these required components.

2. use Android Studio to then install a compatible SDK for React Native. Currently, React Native specifically requires Android 6.0 (Marshmallow) to build apps with native code.

On the *Welcome to Android Studio* screen, the SDK manager may be accessed via the *Configure* menu.

For Android 6.0, React Native specifically requires the following items for build support,

- Google APIs
- Android SDK Platform 23
- Intel x86 Atom_64 System Image
- Google APIs Intel x86 Atom_64 System Image

Then select the tab for *SDK Tools* and check that version 23.0.1 of the Android SDK Build-Tools is selected.

Then, apply these changes and allow Android Studio to download and install these updates.

3. configure environment variables for React Native to work correctly with Android.

We need to update the local `.bash_profile` using the following terminal commands,

```
export ANDROID_HOME=$HOME/Library/Android/sdk
export PATH=$PATH:$ANDROID_HOME/tools
export PATH=$PATH:$ANDROID_HOME/platform-tools
```

We can load this new config for the current session with the following command,

```
source $HOME/.bash_profile
```

and check these were added correctly by using the following command,

```
echo $PATH
```

iOS install

Assuming the initial install outlined above has been completed successfully, including Homebrew, Node.js, Watchman, and React Native CLI, we can then proceed to setup React Native for iOS development.

Xcode needs to be installed, including command line tools. Xcode can be downloaded from the OS X App Store, and further details may be found at the following URL,

- https://developer.apple.com/xcode/

For existing installs, we need to check Xcode is up to date. i.e. all updates must be installed and up to date before proceeding further.

Emulators for various Apple hardware are automatically configured with Xcode and iOS SDK.

Initialise a new app

We can now initialise and create a new React Native app.

At the terminal command line, `cd` to your preferred local directory for development. Then use the following command to create a new app,

```
react-native init FirstApp
```

where `FirstApp` is the name of the new project, and the name assigned to the app's local directory. React Native's CLI tool will now use NPM to download and install a new app, which supports app development for both Android and iOS depending upon local SDK support.

Then `cd` to the new directory, and `ls` the contents to check the install.

test app with Android

To test and run the Android version of a React Native app, we need a working Android device. This may either be a connected device or a local emulator (AVD).

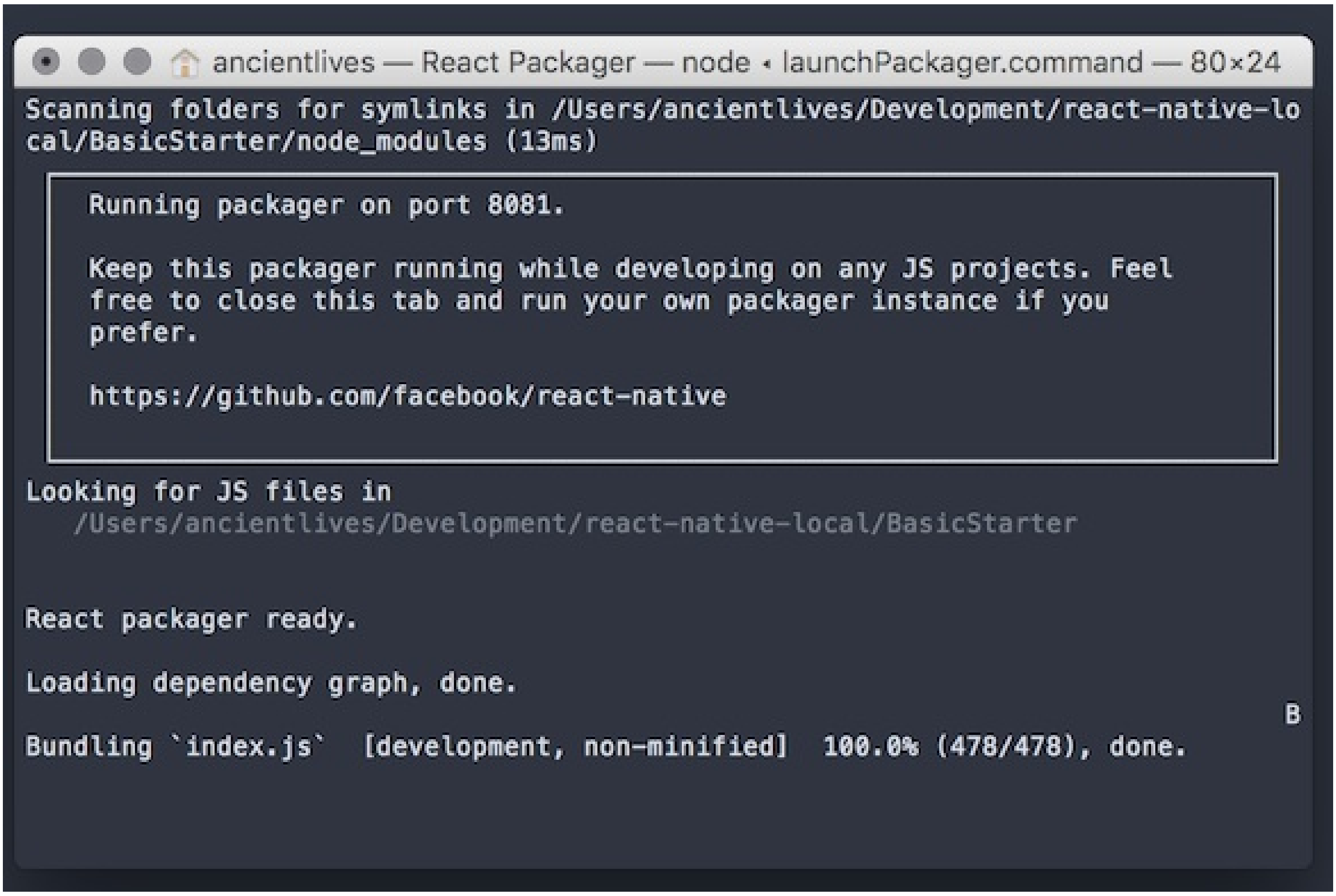
For an emulator, we need to ensure it is up and running before initially launching a local React Native app. In the AVD manager for Android Studio we can simply click the green play button for the emulator supporting API 23.

run local Android app

To run and test a local app, `cd` to the app directory and use the following terminal command.

```
react-native run-android
```

The first time this command is run, the process may take a few minutes to complete. It will also load a JS server, which monitors updates to the local app. For example, the following window will be created,



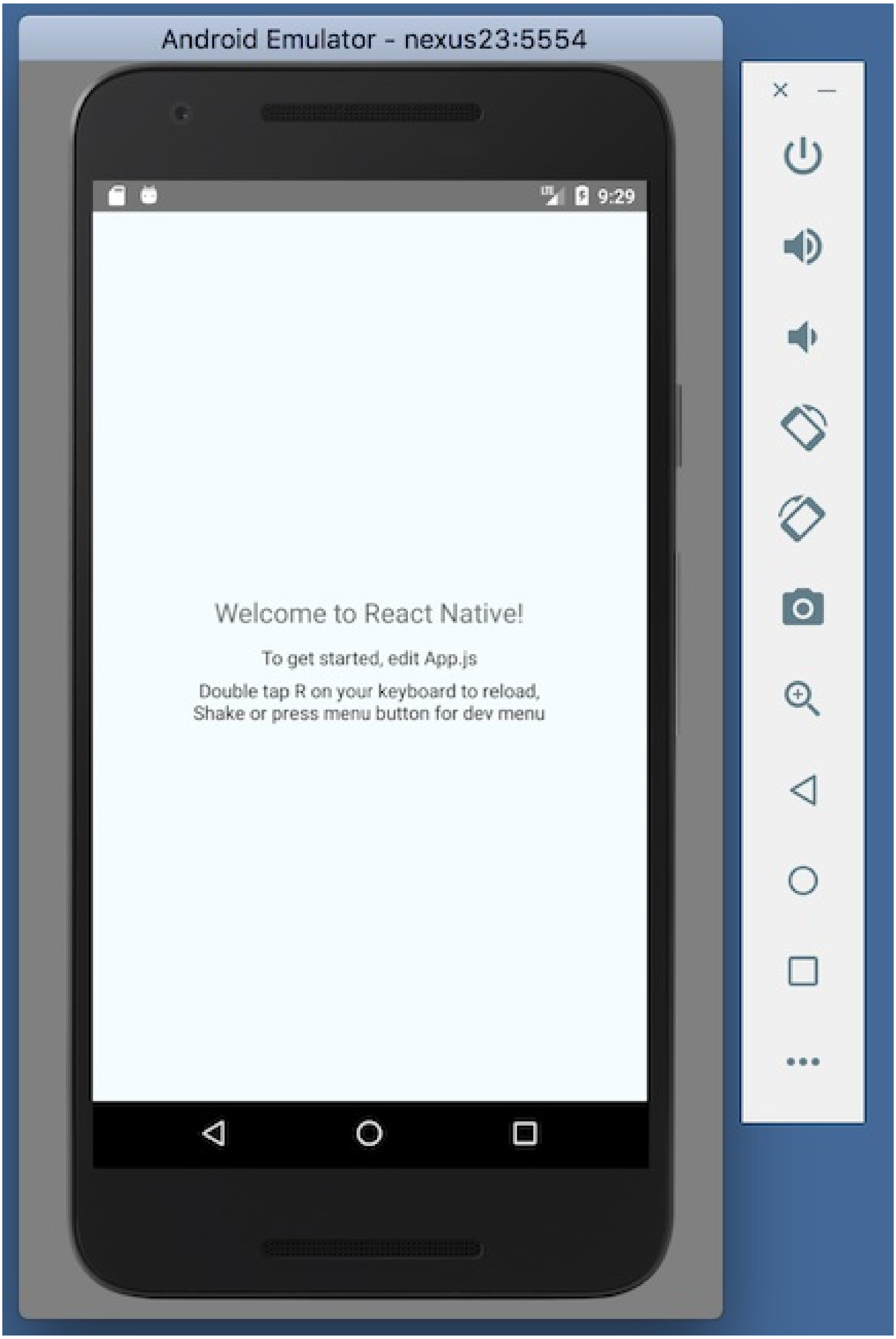
ancientlives — React Packager — node ◂ launchPackager.command — 80×24

```
Scanning folders for symlinks in /Users/ancientlives/Development/react-native-lo
cal/BasicStarter/node_modules (13ms)
 ┌────────────────────────────────────────────────────────────────────────────┐
 │  Running packager on port 8081.                                            │
 │                                                                            │
 │  Keep this packager running while developing on any JS projects. Feel      │
 │  free to close this tab and run your own packager instance if you          │
 │  prefer.                                                                   │
 │                                                                            │
 │  https://github.com/facebook/react-native                                  │
 │                                                                            │
 └────────────────────────────────────────────────────────────────────────────┘
Looking for JS files in
   /Users/ancientlives/Development/react-native-local/BasicStarter


React packager ready.

Loading dependency graph, done.
                                                                               B
Bundling `index.js`  [development, non-minified]  100.0% (478/478), done.
```

The initial React Native app will then load in the running Android emulator,


Android Emulator - nexus23:5554

9:29

Welcome to React Native!

To get started, edit App.js

Double tap R on your keyboard to reload,
Shake or press menu button for dev menu

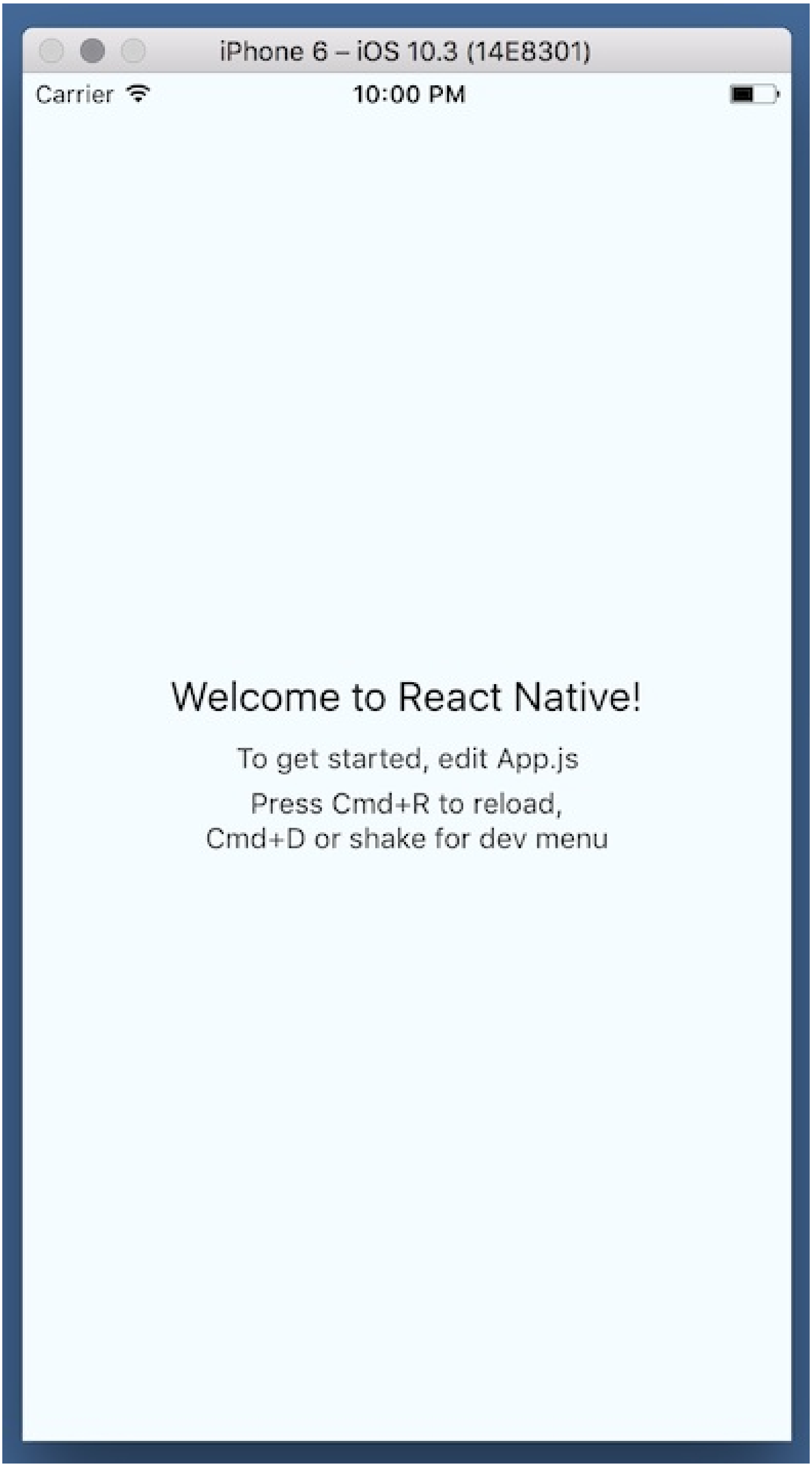
run local iOS app

We can now start our initial project to test that it runs as expected on OS X. We'll need to be in the project directory for the app to load and run. Then, we can issue the following command in the terminal, e.g.

```
react-native run-ios
```

This command will build the project, launch the iOS simulator, and then show the app in a simulator window. For the initial build request, this may take a long time depending upon system performance.

The initial React Native app will then load in the default iOS emulator,



Welcome to React Native!

To get started, edit App.js

Press Cmd+R to reload,
Cmd+D or shake for dev menu

n.b. if the app is not automatically loaded in the simulator window, simply drag the screens in the emulator, and then click on the React Native app icon to launch the app.

terminal commands

Some useful terminal commands for React Native, Android emulator &c.

- launch Android emulator from terminal, e.g.

```
~/Library/Android/sdk/tools/emulator -avd Nexus_23
```