React Native - Install & Config - Windows 10

- Dr Nick Hayward

Install and configure React Native on Windows 10 for Android development.

Contents

- intro
- basic dependencies
- React Native CLI
- Android install
- initialise a new app
  - test app with Android
  - run local Android app

Intro

React Native on Windows 10 includes support for native development with Android.

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 Windows 10 options, simply select the button for *Windows*, and *Android* to see further details.

basic dependencies

Basic dependencies include,

- Node.js
- Python 2.x
- JDK (Java SE Development Kit)
- Android Studio

React Native itself needs Node.js installed,

- https://nodejs.org/en/

As with Mac OS X, there are package manager options we can use to install, monitor, and update many of these required packages.

Chocolatey, which is based on the internal NuGet package manager used by the Windows development team, offers a quick and easy option for package management. To install *Chocolatey*, follow the instructions at the following URL,

- Install guide

Choose either *Install with cmd.exe* or *Install with PowerShell.exe* to correctly download and configure Chocolatey.

If you install Chocolatey with Powershell, you'll need to use the *Adminstrator* option and check the *execution policy* for your local system. Once this has been modified, as detailed in the install instructions for Chocolatey, you'll be able to install the package manager and associated software.

Then, use the following command to install Node.js, Python 2.x, and JDK with Chocolatey,

```
choco install -y nodejs.install python2 jdk8
```

After Chocolatey has finished installing these packages, you'll need to then restart Powershell.

We can then check install versions using the following command,

```
java -version
node -v
python --version
```

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 create a new environment variable for `ANDROID_HOME` using the following steps,

- open *Settings* and type *System Info* in the search box
- select the result for *System*
  - select the new window *Control Panel\System and Security\System*
  - select the link *Advanced system settings* in the left menu
  - in the *System Properties* window select the button *Environment Variables* in the bottom right corner
  - in the *Environment Variables* window select the *New* button for *User Variables*
  - set the new variable name to *ANDROID_HOME* and the value to the location of the SDK, e.g.
    - `C:\Users\your_username\AppData\Local\Android\sdk`

Initialise a new app

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

Using Powershell, for example, `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).

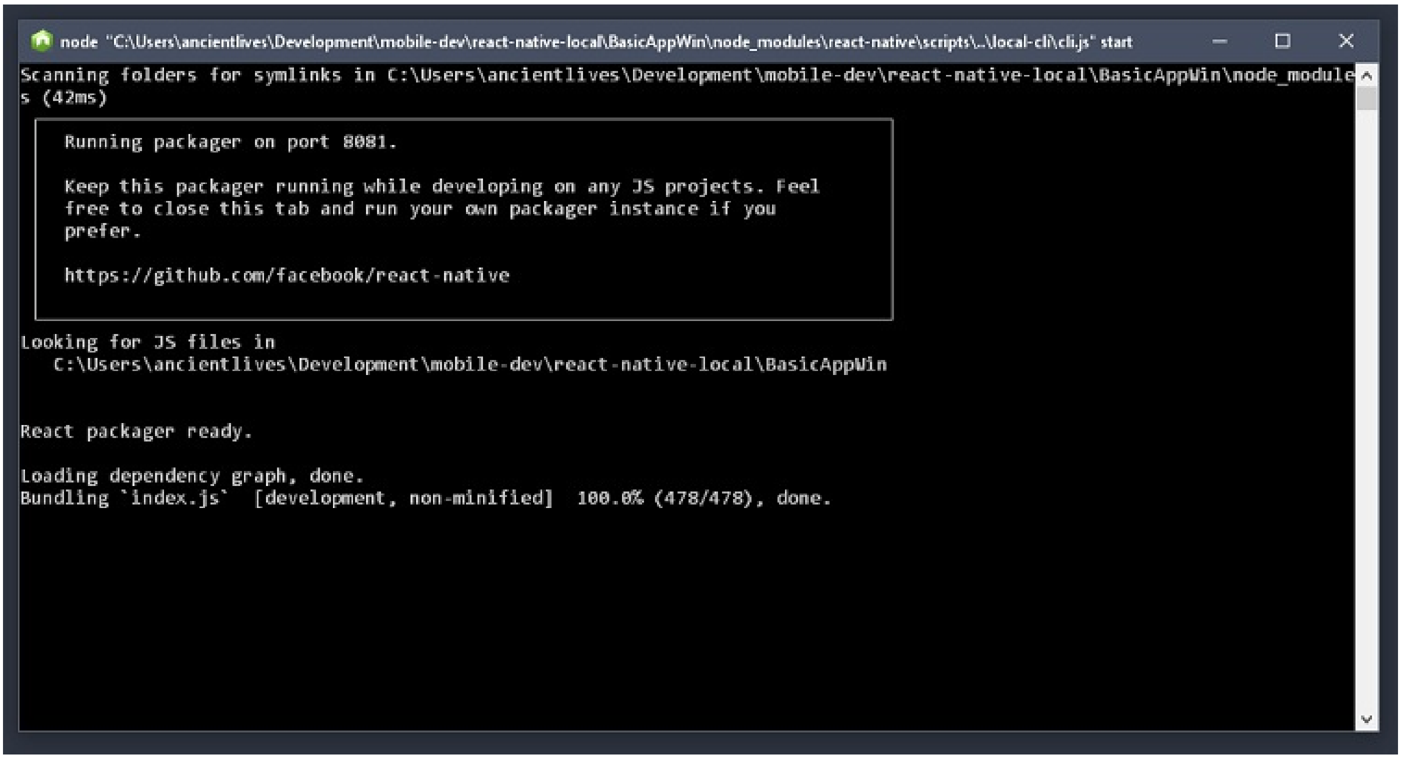
For an emulator, we need to ensure it is up and running before initially launching a local React Native app. In the AVD manager for Android Studio we can simply click the green play button for the emulator supporting API 23.

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,



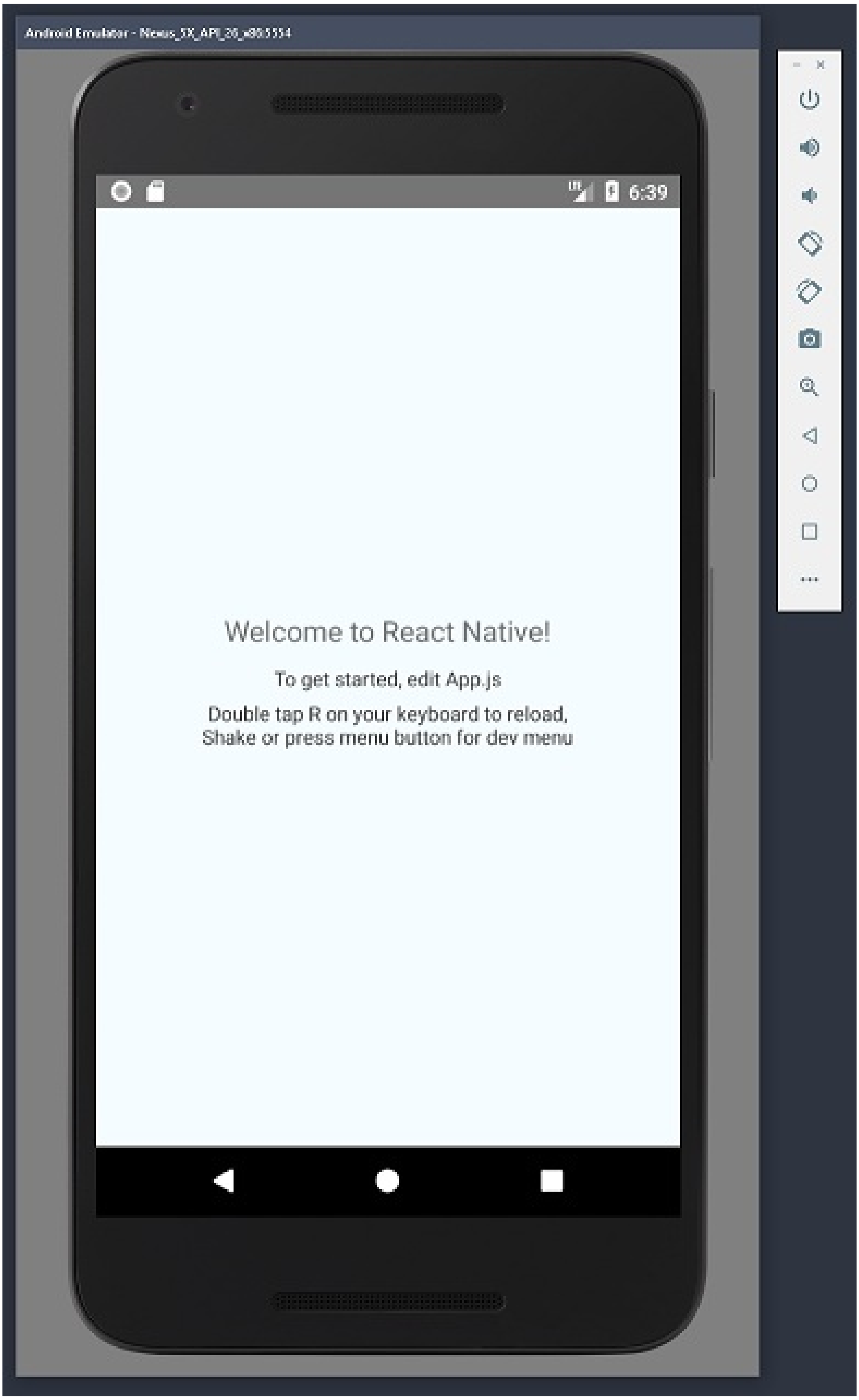
node "C:\Users\ancientlives\Development\mobile-dev\react-native-local\BasicAppWin\node_modules\react-native\scripts\..\local-cli\cli.js" start

```
Scanning folders for symlinks in C:\Users\ancientlives\Development\mobile-dev\react-native-local\BasicAppWin\node_module
s (42ms)
 ┌──────────────────────────────────────────────────────────────────────────┐
 │  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
   C:\Users\ancientlives\Development\mobile-dev\react-native-local\BasicAppWin


React packager ready.

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

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



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