Lab 1: Environment Setup for Windows

Adapted from http://facebook.github.io/react-native/docs/getting-started.html. Follow these steps to configure a working development environment on your computer.

In this exercise, you will:

- Install Windows development environment dependencies
- Install Android Studio and the Android 6.0 (Marshmallow) SDK
- Configure your device for development
- Install an IDE
- Create and configure a Facebook application in the Application Dashboard

The time required to complete this exercise will vary depending on the speed of your network connection and the state of your computer. You should expect to spend a minimum of one hour on these tasks.

Install Windows Development Environment Dependencies

1. Bookmark this URL in your browser: http://bit.ly/MasterclassForDevelopersStudentFiles

---

Note: This location will be known as {student-files} throughout this document.

---

2. Open an elevated command prompt and run the following command to install Chocolatey, a package manager for Windows:

```
@powershell
-NoProfile -ExecutionPolicy Bypass -Command "iex ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps
1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"\
```

Note: this command was taken from https://chocolatey.org/install#install-with-cmdexe and is listed in *{student-files}\commands.txt* for your convenience. If you encounter problems with this step, consult that site for other options.

---

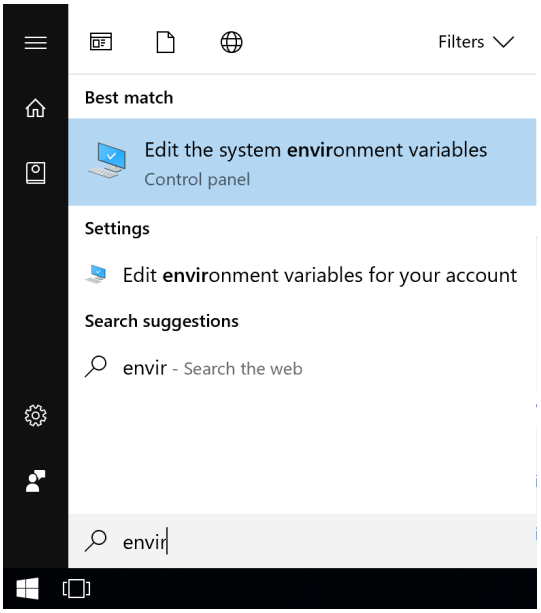
3. After the Chocolatey installation is finished, run each of the following commands, one after the other, in the elevated command prompt.

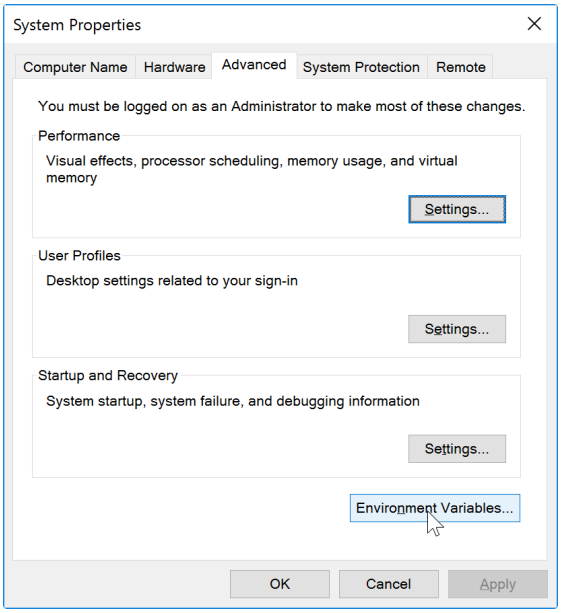
```
choco install nodejs.install
choco install python2
choco install jdk8
choco install yarn
choco install git
```

Note: respond with 'y' each time you are prompted to run the installation script.

---

4. Open the Environment Variable editor.





Note: You may use the shortcut illustrated above (, envir) or follow Control Panel → System and Security → System → Change settings → Advanced tab → Environment Variables button.

---

5. Ensure the following additions have been made:
    a. **User** Path variable:
        - C:\Users\{username}\AppData\Roaming\npm
        - C:\Users\{username}\AppData\Local\Yarn\.bin
    b. **System** Path variable:
        - C:\Program Files\nodejs\
        - C:\Python27
        - C:\Program Files\Java\jdk1.8.0_{xxx}\bin
        - C:\Program Files\Yarn\bin
        - C:\Program Files\Git\cmd

Note: Most of the environment variable changes listed above should have been made automatically by the installation scripts. If they have not been added, you must add them now.

---

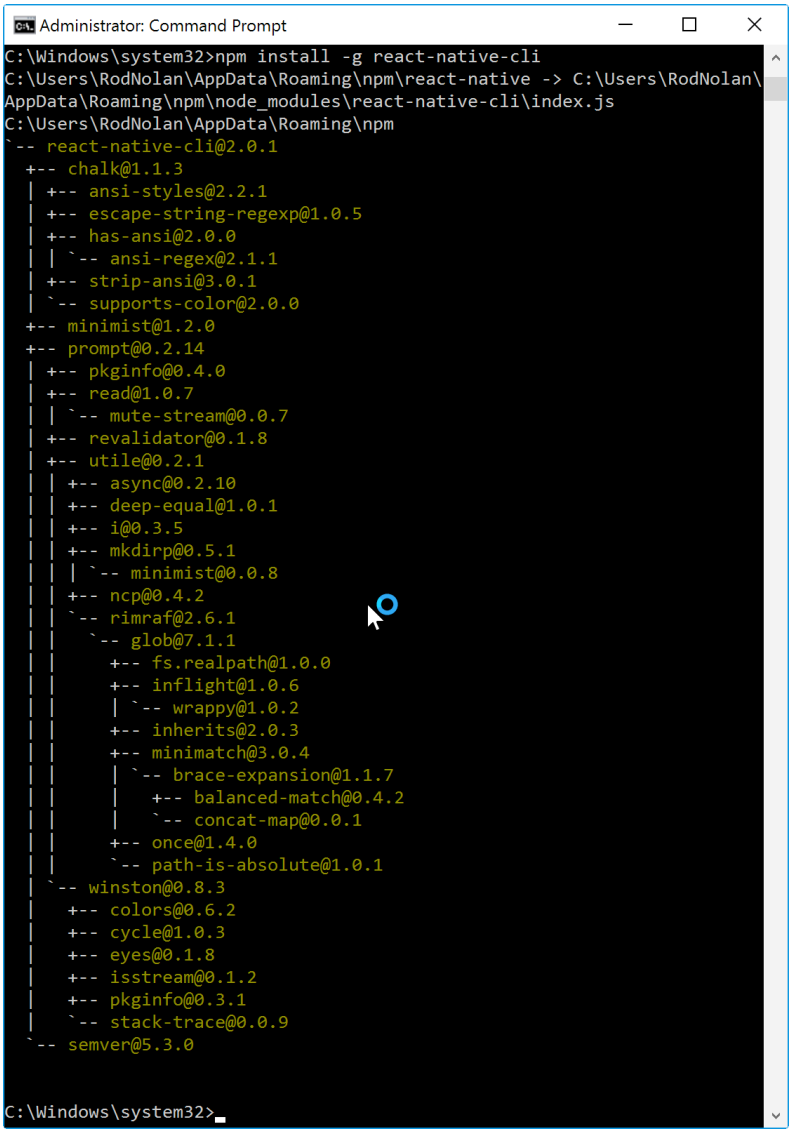
6. Add a new **System** variable:
        - JAVA_HOME = C:\Program Files\Java\jdk1.8.0_{xxx}

Note: Update the '{xxx}'part of the value above to match the version number that was installed on your system. Open C:\Program Files\Java\ to check the folder name.

Install the react-native command line interface using npm

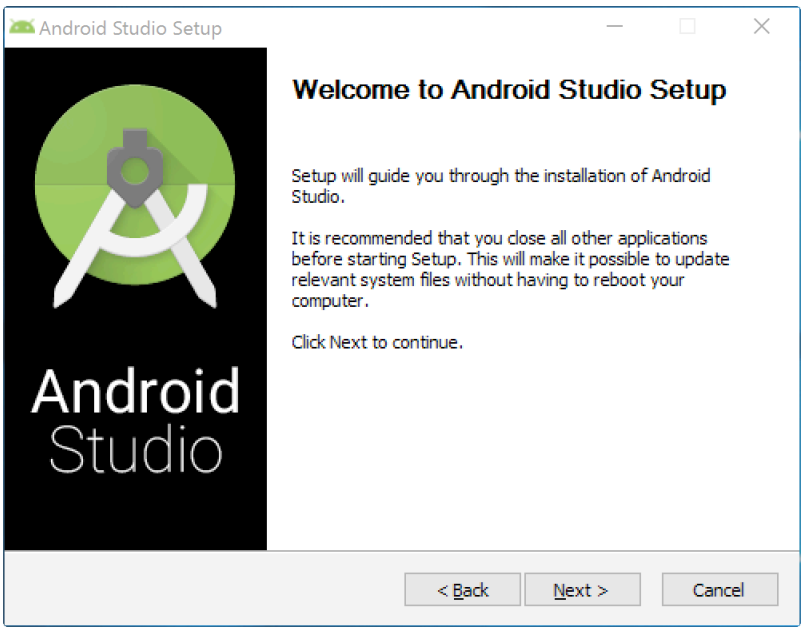
7. Use an elevated command prompt (⊞ + X, A) to install the react native command line interface.

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

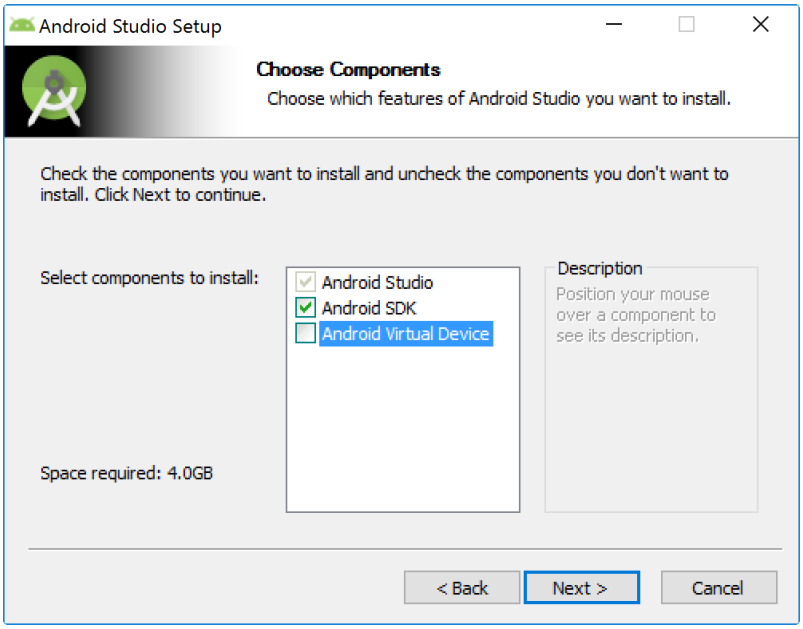


Install Android Studio

8. Download and run the latest Android Studio installer from https://developer.android.com/studio/index.html.



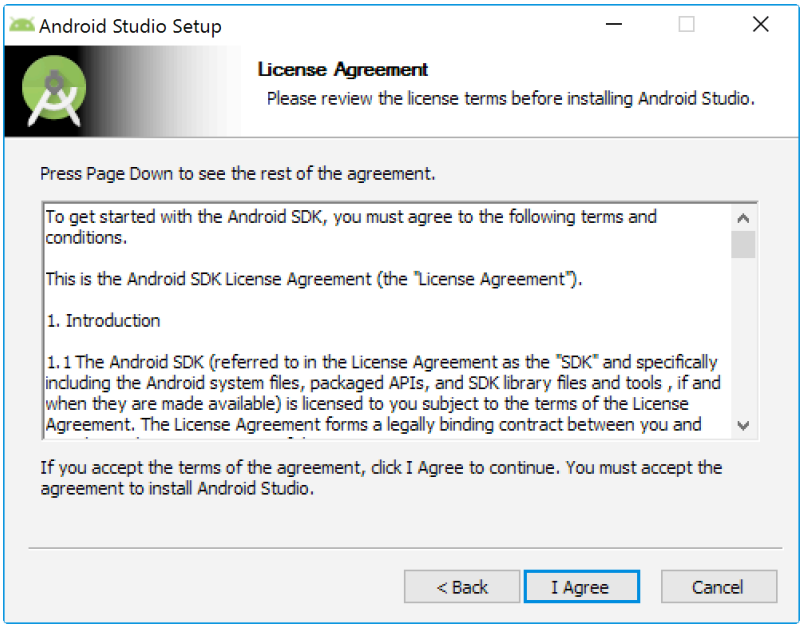
9. Uncheck the Android Virtual Device option.



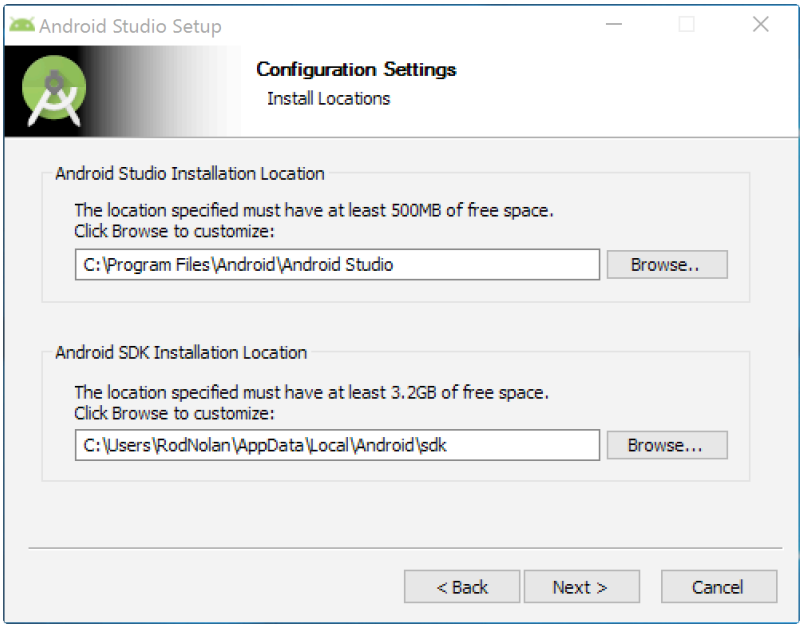
Note: The Android Virtual Device is only required to use the emulator. If you have a physical Android device you can save time and disk space by deselecting this option. Otherwise, check this option if you must use the emulator.

---

10. Agree to the License Agreement

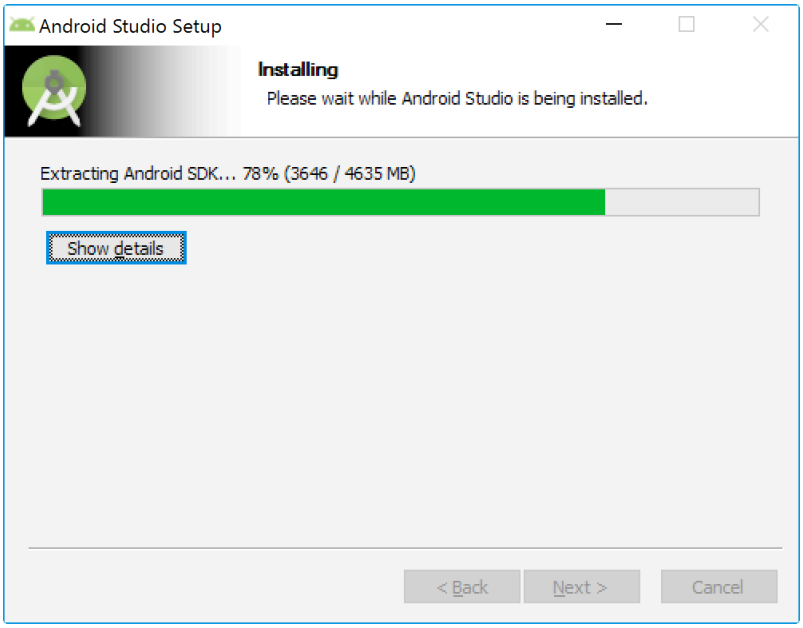


11. Accept the default installation paths



Note: Take note of this location on your system. You will need to refer to it in a later step.

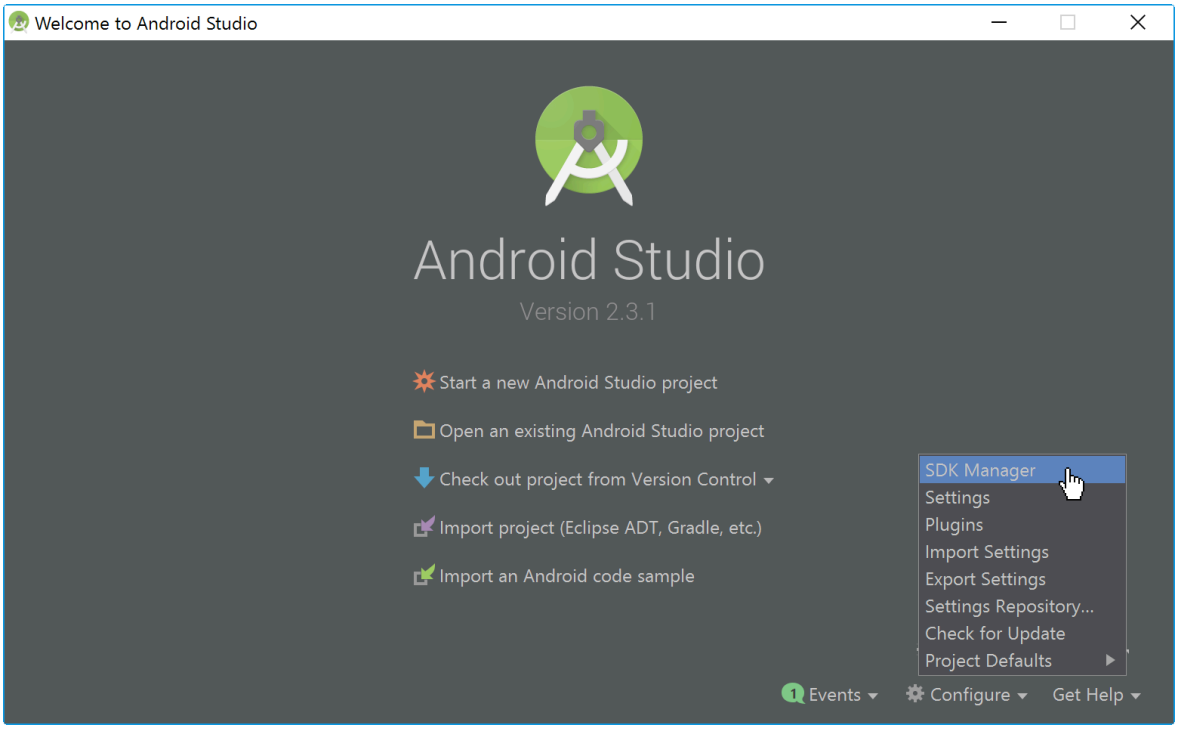
12. Wait until the installation finishes.



Note: The Extracting Android SDK step takes a long time so be patient.

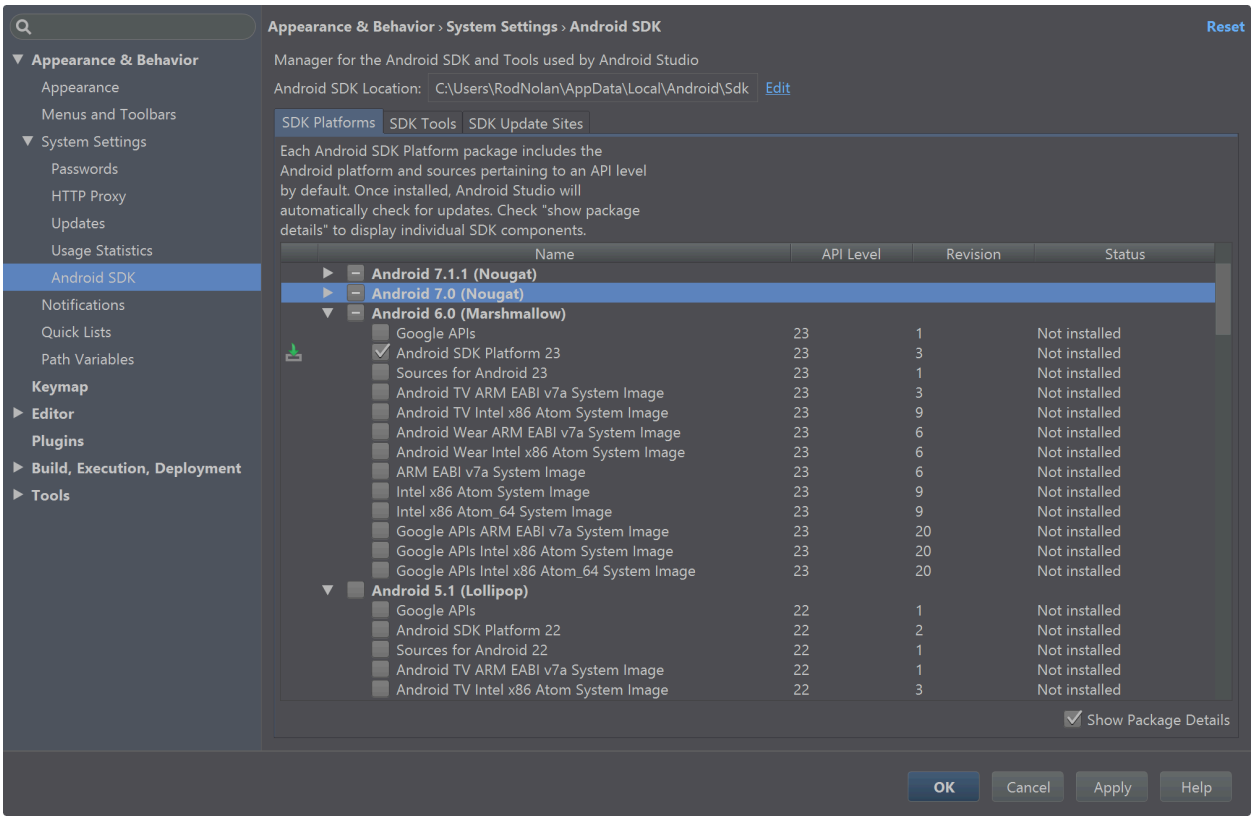
Install the Marshmallow SDK

13. Start Android Studio and open the SDK Manager



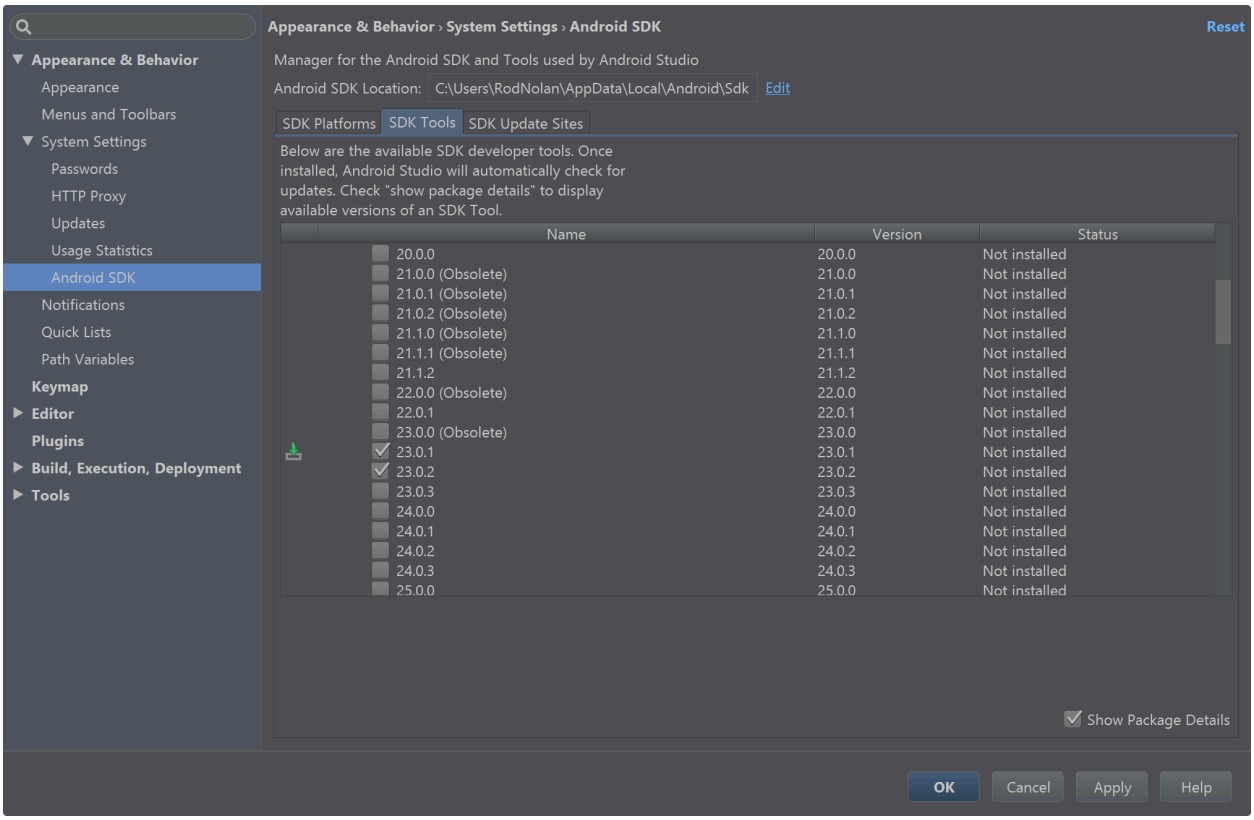
Note: The Android Studio installer installs the most recent SDK by default. However, React Native requires the Android 6.0 Marshmallow SDK. The SDK Manager is used to manage installed SDKs.

14. Install the Android SDK Platform 23
    a. Click the Show Package Details checkbox in the bottom right corner of the screen
    b. Select the Android SDK Platform 23

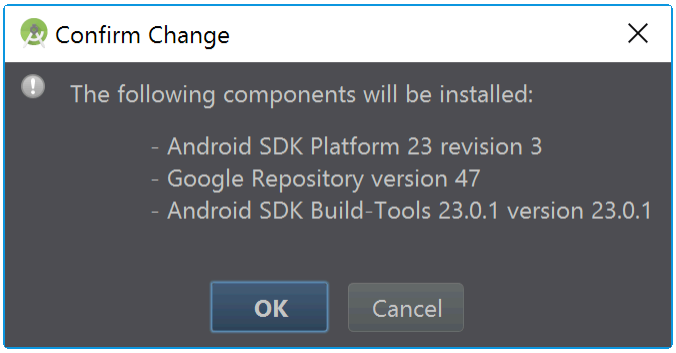


Note: The *Google APIs*, *Intel x86 Atom_64 System Image* and *Google APIs Intel x86 Atom_64 System Image* options are only required to run the emulator. A real device is required for much of this course so you can save setup time by deselecting the emulator-related options now.

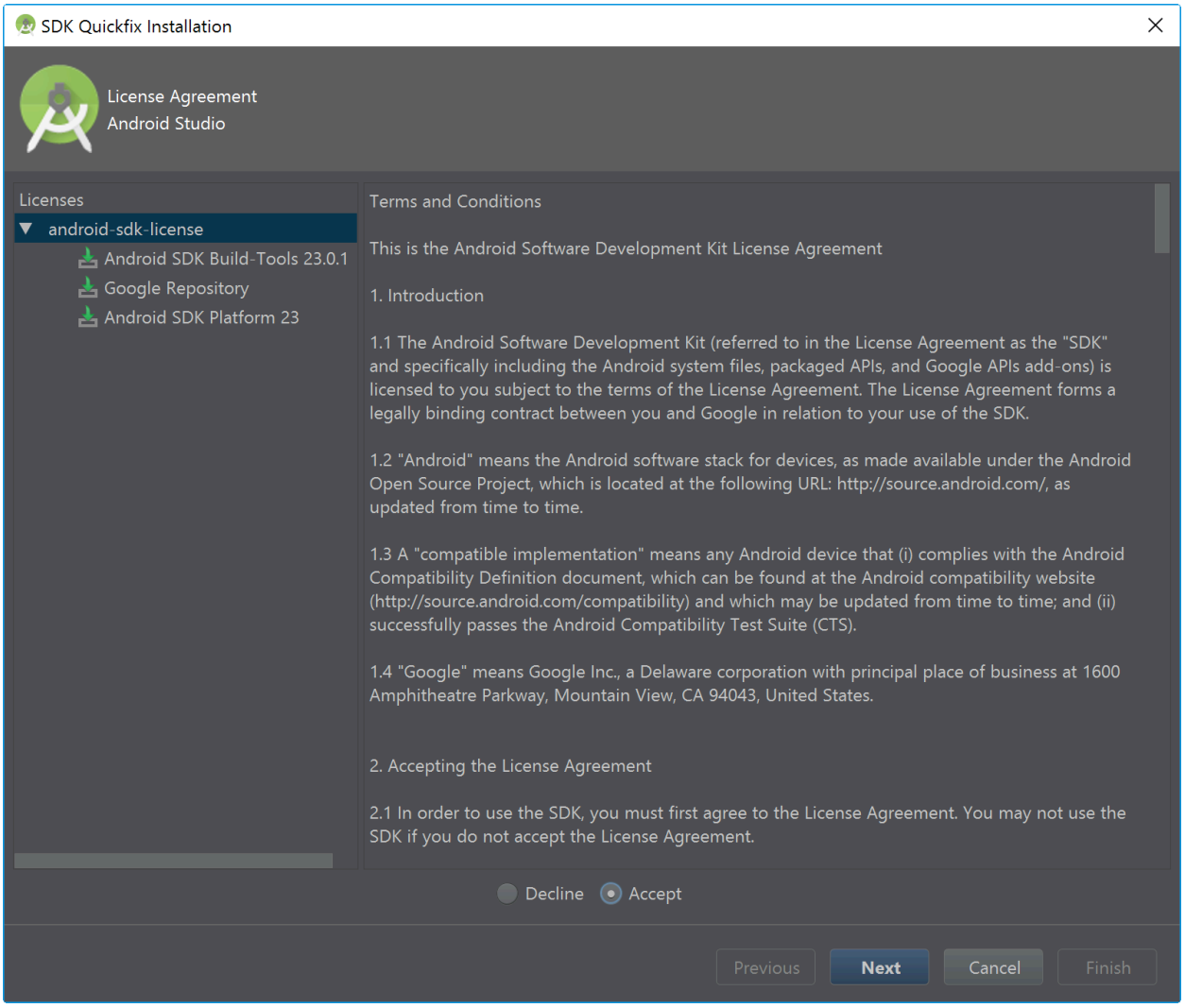
15. Install the Marshmallow build tools
    a. Switch to the SDK Tools tab
    b. Click the Show Package Details checkbox in the bottom right corner
    c. Expand the Android SDK Build Tools category
    d. Select 23.0.1 and 23.0.2 (23.0.2 facilitates faster builds by running dex in-process)
    e. Click the Apply button



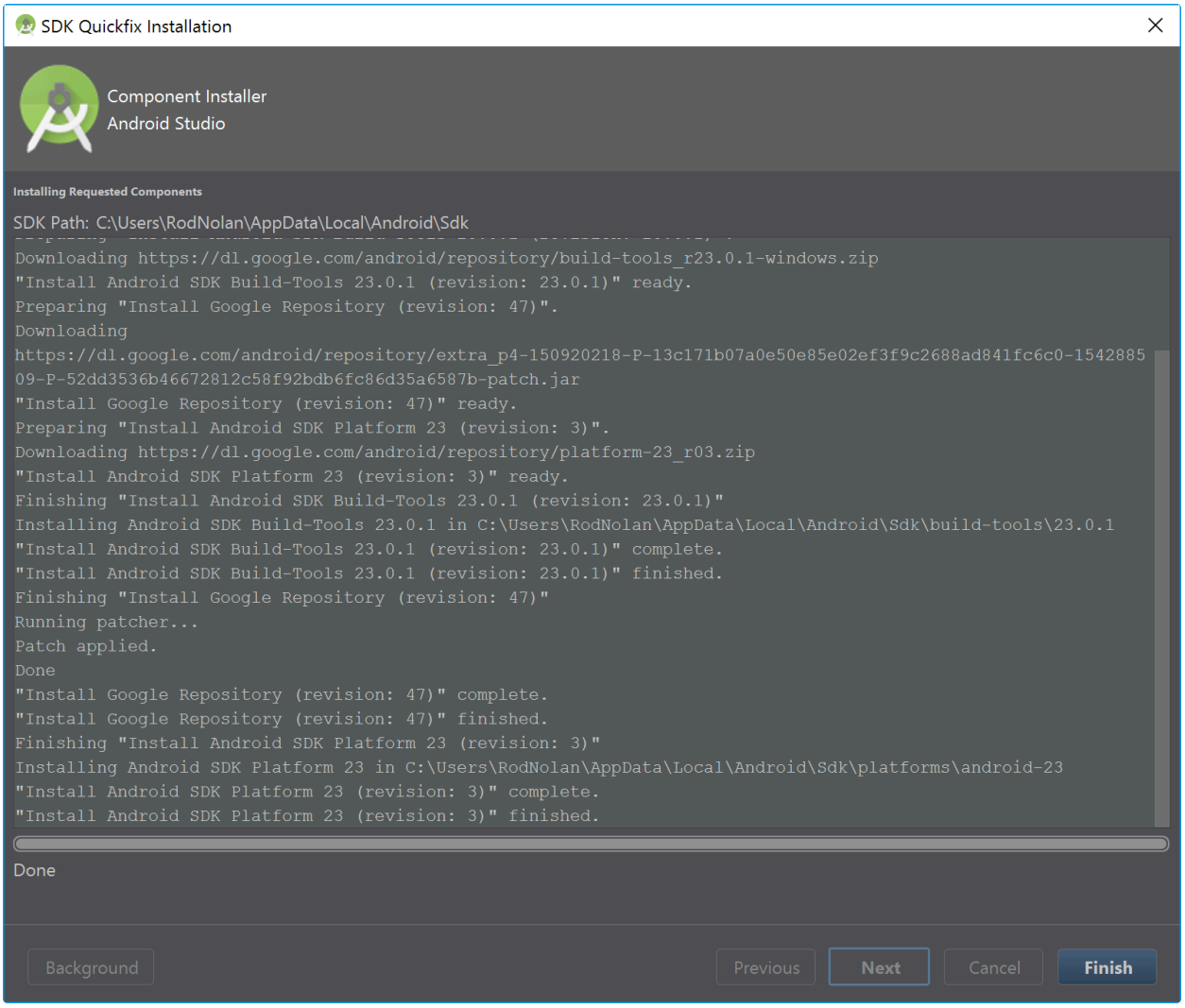
16. Confirm the changes.



17. Accept the terms and conditions and click the Next button to start the installation.

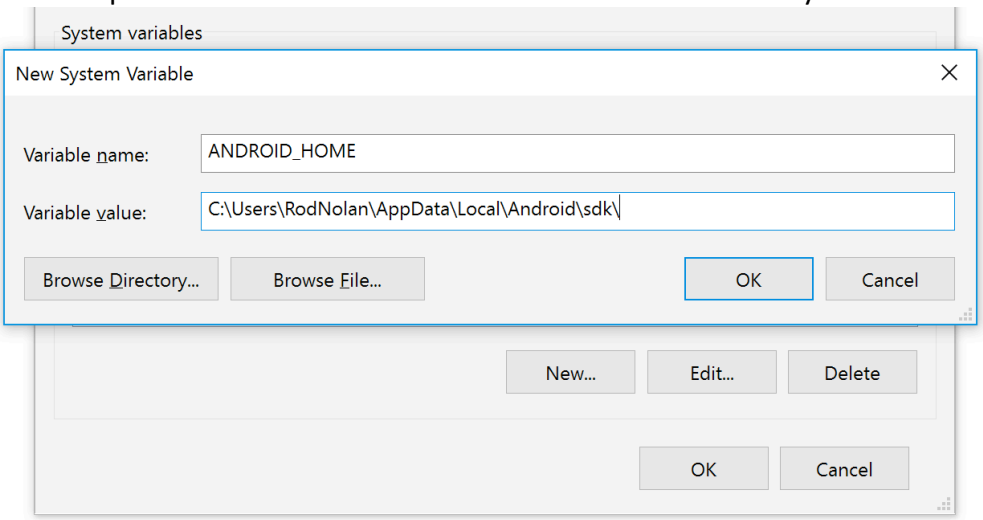


18. When the installation finishes, click the Finish button.



Set the ANDROID_HOME environment variable

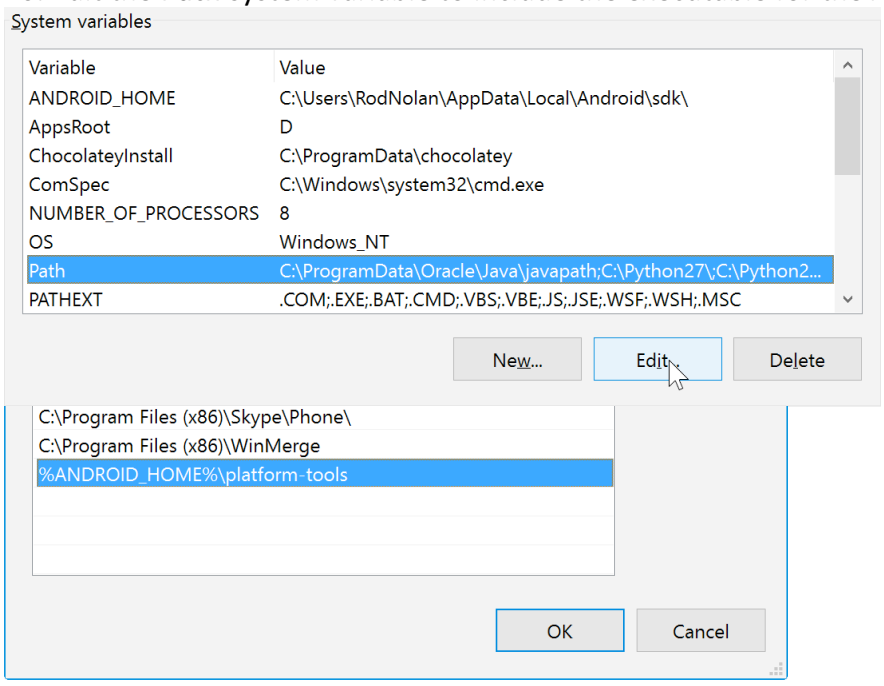
19. Open the environment variable editor and add a new System Variable



Note: Update the path listed above to match the one for your own system or use the following: *%USERPROFILE%\AppData\Local\Android\sdk*

---

20. Edit the **Path** system variable to include the executable for the Android Debug Bridge.

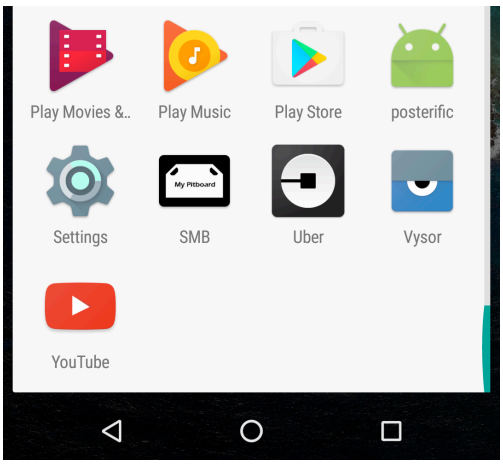


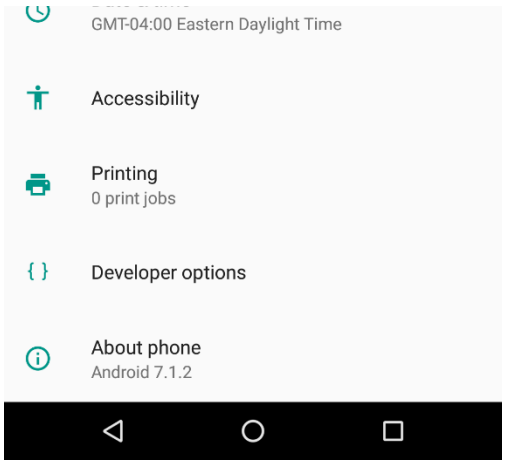
Note: adb.exe is located in the platform-tools folder – this is required to debug on an Android device using the react native cli

---

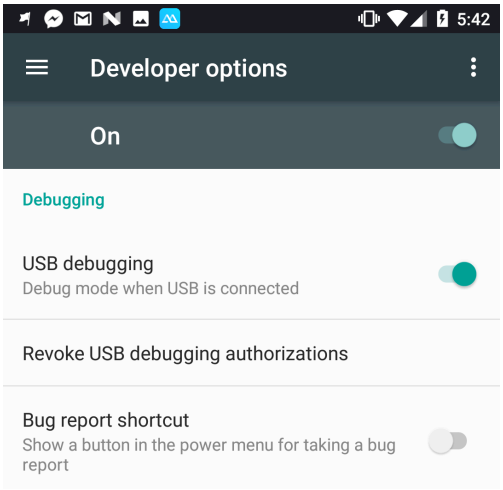
Enable USB debugging on your device

21. Open Settings → Developer Options





22. Turn the Developer options switch on (top of the screen) and then scroll down to the Debugging section and turn on the USB debugging switch.



23. Connect your device to your computer with a USB cable.

Use the react native cli to create a new project

24. Open a command window and enter the following commands:

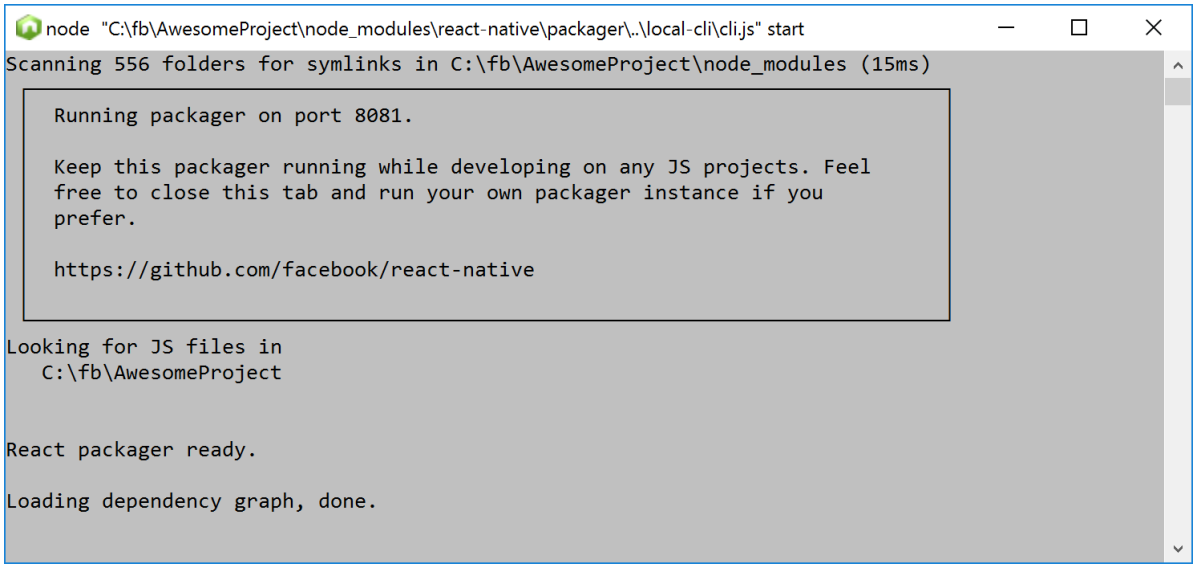
```
cd \
mkdir fb
cd fb
react-native init AwesomeProject
cd AwesomeProject
react-native run-android
```

Note: The first time you do this, some additional tools related to the Android build system will be downloaded. This is a one-time operation but it can take a considerable amount of time.

---

25. The React Native Packager will open automatically in a separate command window. You may also open it manually, in advance, with the following command:

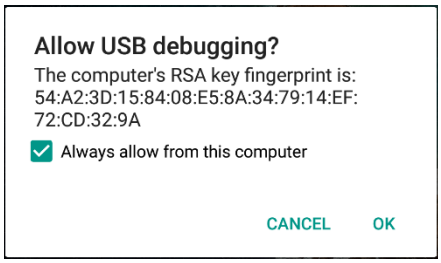
```
react-native start
```



Note: This window should remain open while you are editing your application's source code.

---

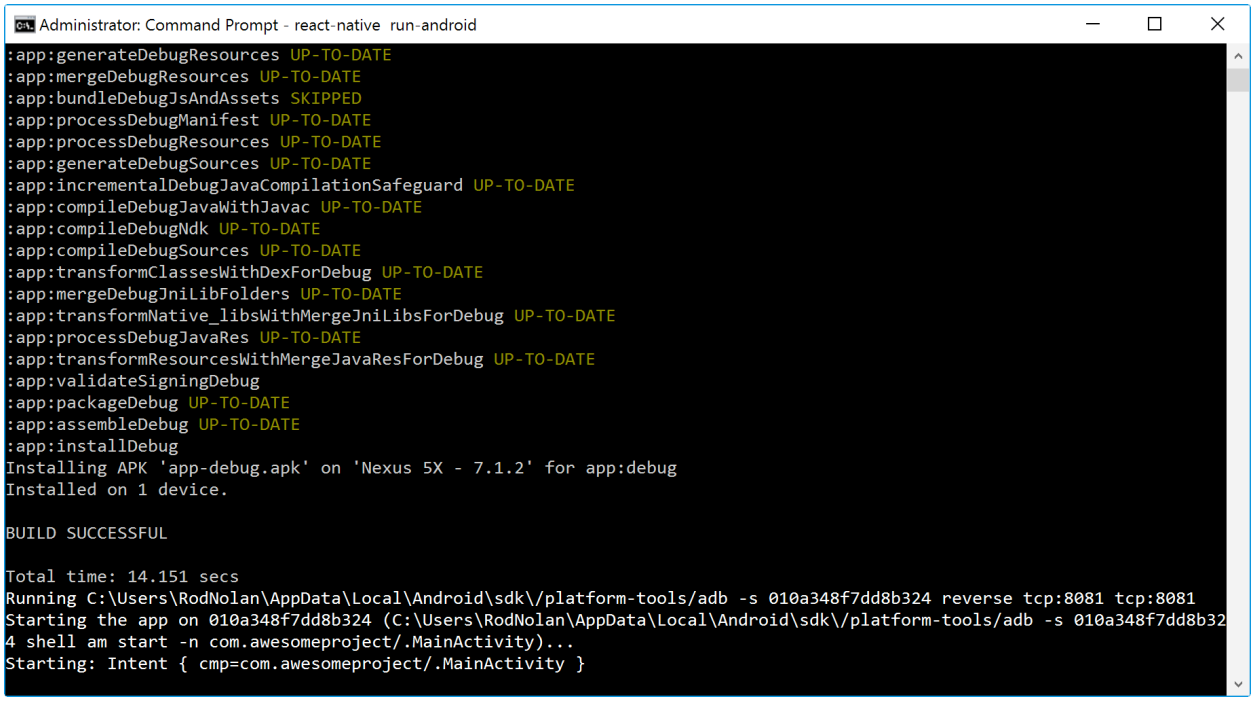
26. Near the end of the build process your device will prompt you to Allow USB Debugging on your computer. This prompt will only appear if the device is unlocked. This prevents unauthorized access to the device by requiring the user to enter the unlock code. Check the Always allow from this computer option and click OK.



Note: If you must install USB drivers, see the instructions listed here:
https://developer.android.com/studio/run/oem-usb.html
https://developer.android.com/training/basics/firstapp/running-app.html

---

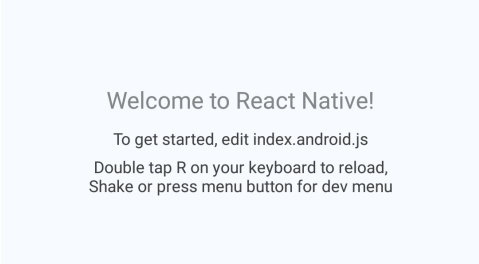
27. The application should build and install as illustrated below.



Note: If this build attempt fails, you will see an error message in red in the console. Address the issue and try running the *react-native run-android* command again.

---

28. The application should open on your device.



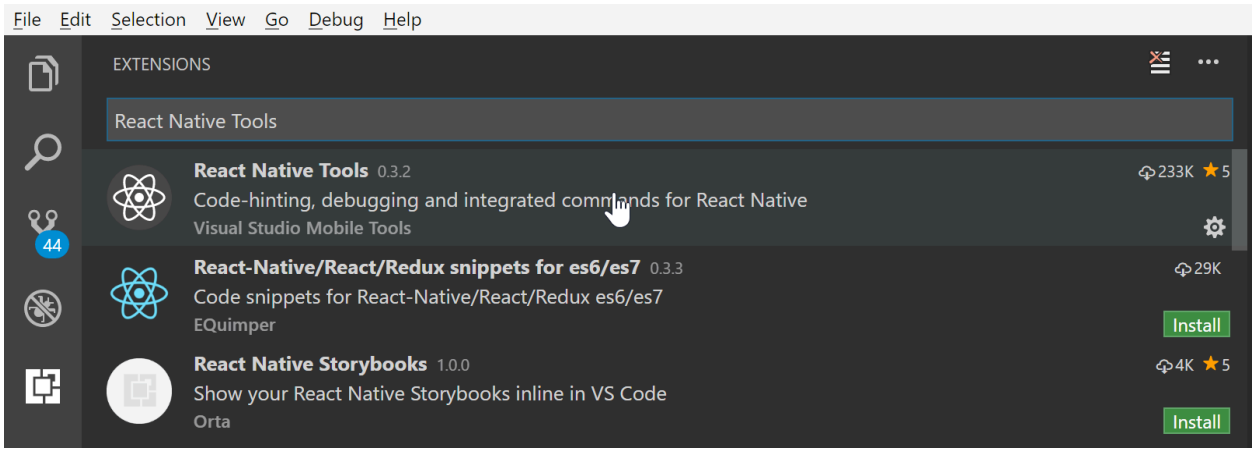
Install an IDE

29. Download and run the latest Visual Studio Code installer from https://code.visualstudio.com.

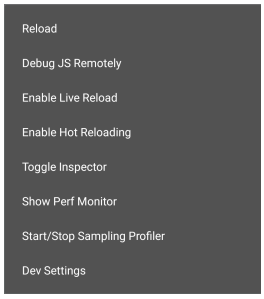
Note: Choosing an IDE is a matter of personal preference. After experimenting with various editors, the course authors settled on Visual Studio Code. Feel free to install an IDE that more closely matches your preferences after you finish this exercise.

---

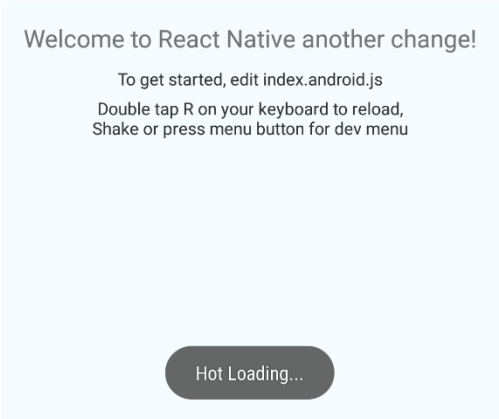
30. Install the React Native Tools extension.



31. In VSCode, choose File → Open Folder and open *C:\fb\AwesomeProject.*
32. Open the file *index.android.js*.
33. Modify the line of text that reads Welcome to React Native! and save the file.
34. Shake the device to invoke the dev menu and select Reload to see your changes.

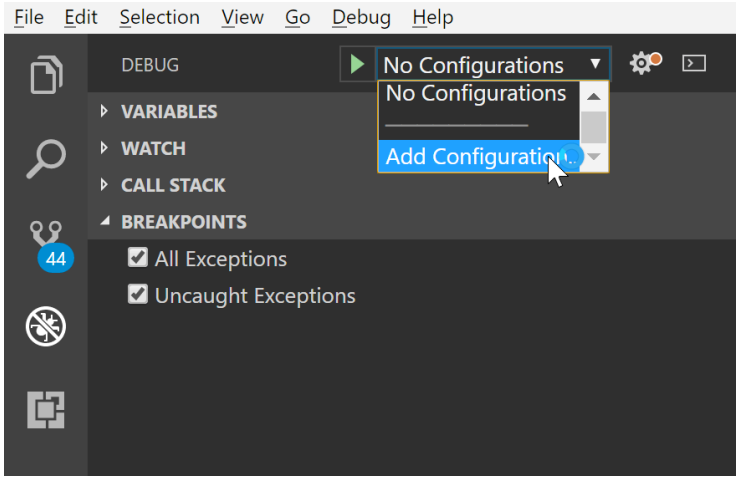


35. You can optionally select Enable Hot Reloading so that your changes are automatically reloaded whenever you save a .js file in your project's source folder.



Add a launch configuration for debugging on Android

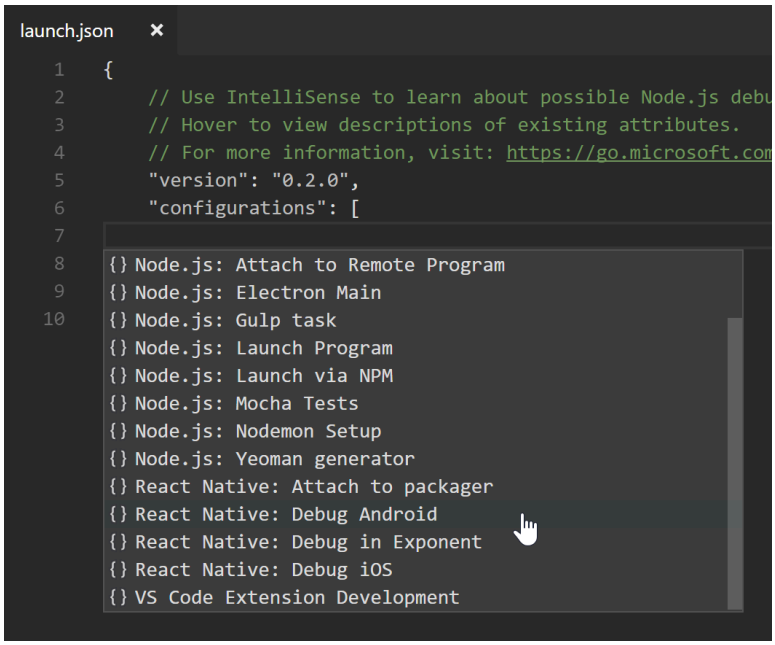
36. Switch to the Debug pane in Visual Studio Code, open the No Configurations dropdown and select Add Configuration.



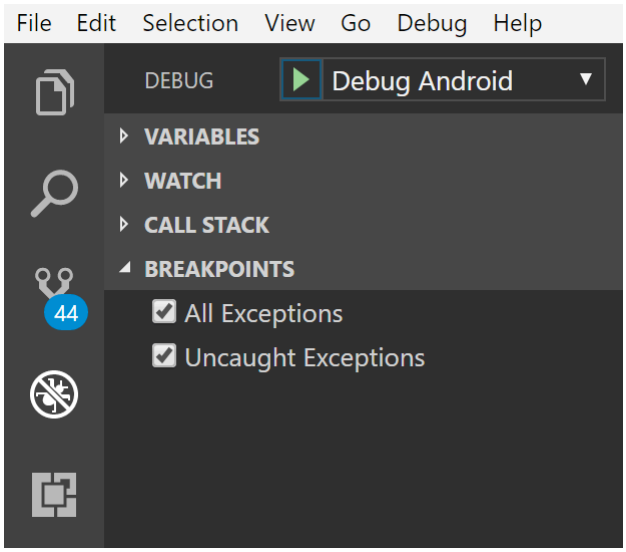
Note: This will create a new file in your workspace (.vscode\launch.json) and open it for editing.

---

37. Select the React Native: Debug Android option.



38. Note the new Debug Android option.



39. Close the two Command windows you opened earlier and close the app on the device.

Note: Another instance of the react native packager will be opened when you complete the next step and only one instance can be open at any given time.

---

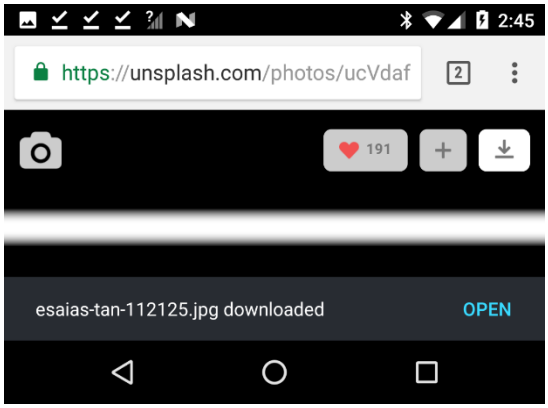
40. Click the green play button with the Debug Android run configuration selected to start a new debugging session from the IDE.

41. Note that the debugging output appears in the Debug Console and the debugging controller appears near the top of the editor window.

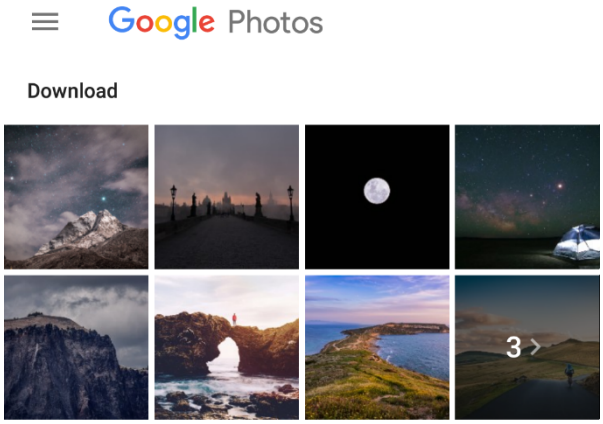
42. Experiment by changing some of the text and watch the effect of the code changes appear on your device. Click the stop button in the debugger when you are finished.

Download some images to your device

43. Open a browser on your device and visit https://unsplash.com
44. Browse the collections and save a few images.



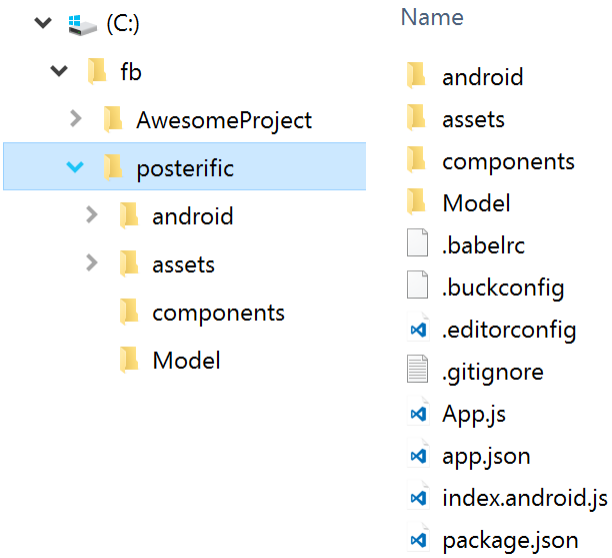
45. Open the Google Photos app and go to the Download category to verify that the images were saved.



Note: The images you saved from the browser should be stored in the device's Download folder. The sample app that you will work with will look in this folder for images.

Setup the Posterific project in your editor

46. Extract the contents of the *{student-files}\the-posterific-app\basic.zip* archive to *C:\fb*. Your folders should look like this:



47. Open the *C:\fb\posterific* folder in your editor.
48. Open and explore the following files:
    a. package.json: defines project dependencies
    b. App.js: the application's entry point; imports all screens and configures routing
    c. node_modules/: stores downloaded copies of modules specified in package.json
    d. Model/: value objects and other data storage-related classes
    e. assets/images/: images for icons, buttons and screen backgrounds
49. Explore the android/ folder:
    a. settings.gradle: defines additional libraries to compile
    b. app/build.gradle: defines additional project dependencies (see dependencies section)

Note: These two files are usually modified automatically by the React Native CLI but there may be times when you will need to update them manually.

---

    c. app/src/main/AndroidManifext.xml: defines project configuration settings
    d. app/src/main/res/values/strings.xml: defines variables used in AndroidManifest.xml
    e. app/src/main/java/com/posterific/: you will modify these files are you integrate external libraries into the project
    f. build.gradle: You should not need to edit this file

50. Open a command prompt and install these dependencies.

```
yarn install
```

Note: All dependencies will be installed in the *node_modules* folder.

Run the app on the device

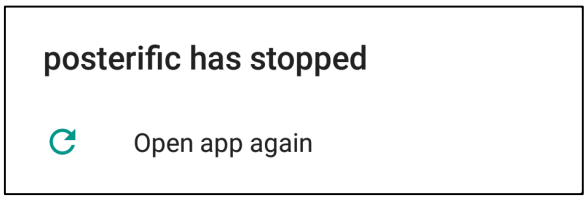
51. Ensure that your device is connected to your computer via the USB cable.
52. Compile and install the posterific app on the device with the following command.

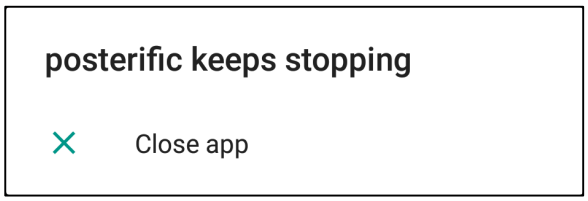
```
react-native run-android
```

Note: You could also use the Debug Android run configuration in Visual Studio Code. If Debug Android is not in the list, select Add Configuration… to create it.

---

53. The application may crash while it attempts to connect to the packager. If it does, you may have to open the application again or close and re-open it manually.





Use the application to familiarize yourself with its screens.

54. Click the Get Started button to navigate to the *Your Saved Posters* screen. Note the following:
    a. Click the add button in the bottom right corner to create a new poster.
    b. If you have created at least one poster, click the button in the top right corner to toggle between list view and grid view.
    c. Three actions can be taken on each item:
        i. click the poster's image to load it back into the designer.
        ii. click the Cart button to begin the checkout process.
        iii. click the Delete button to remove the item from the list.
55. On the *Design Your Poster* screen, there are six active buttons:
    a. Click Rotate to toggle between landscape and portrait orientation.
    b. Click Photo to select one of the photos from your Downloads folder.
    c. Click Caption to modify the caption that appears at the bottom of the poster.
    d. Click Background to change the color of the photo matte.
    e. Click the Save button to save the poster as it is currently configured and return to the *Your Saved Posters* screen.

f. Click the Buy Now button to auto save the poster and begin the checkout process.

Note: The Messenger button (middle, bottom) is not yet active.

---

56. The *Order Confirmation* screen allows you to review your creation. There are two ways to proceed:
    a. Click the Back button at the top left to navigate back to the previous screen.
    b. Click the Proceed to Checkout button to “finalize” your order.

Note: This does not integrate with a real transaction system.

---

57. The Checkout screen provides two buttons.
    a. Click the Back button at the top left to navigate back to the previous screen.
    b. Click the Buy Now button to see a toast notification indicating success.

Sign up as a Facebook Developer and create a new application

58. Login to Facebook in a desktop browser.
59. Visit https://developers.facebook.com/
60. Click the Get Started button and follow the prompts as illustrated below.

Note: If you are already a registered Facebook Developer, you don't have to register again. You can skip this step and start on the next page.

---

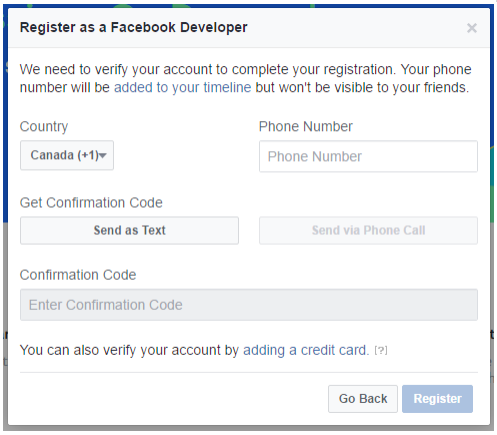
a. Accept the Platform and Privacy policies.

b. If necessary, look up your phone number.

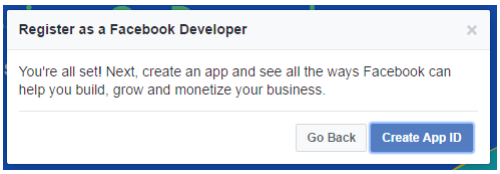
Note: Settings → About phone → Status → SIM status → My phone number

---

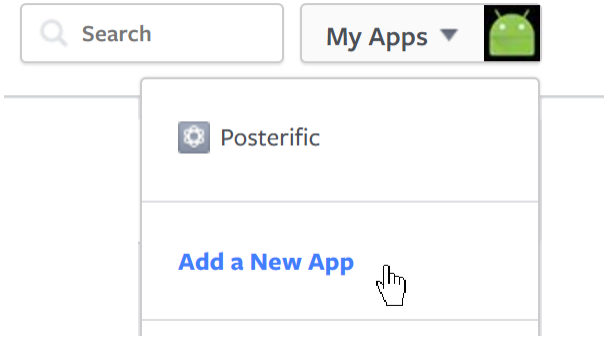
c. Enter your mobile number and click the Send as Text button or the Send via Phone Call button.
d. Enter the Confirmation Code that you receive and click the Register button.



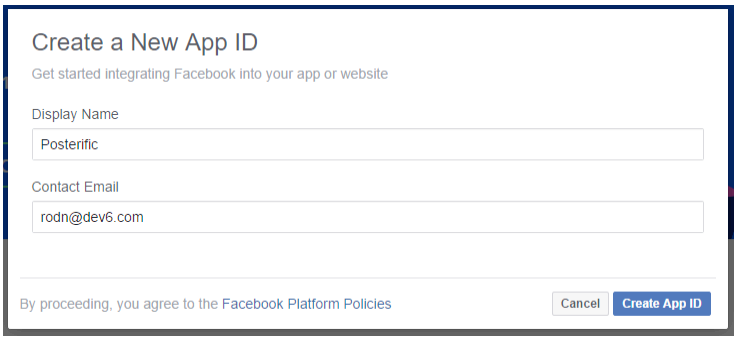
61. Click the Create App ID button.
    a. If you are registering for the first time, you will see this.



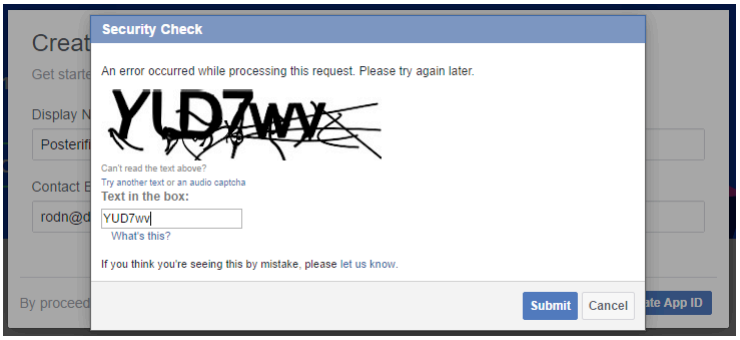
    b. If you are already a registered developer, you can Add a new App.



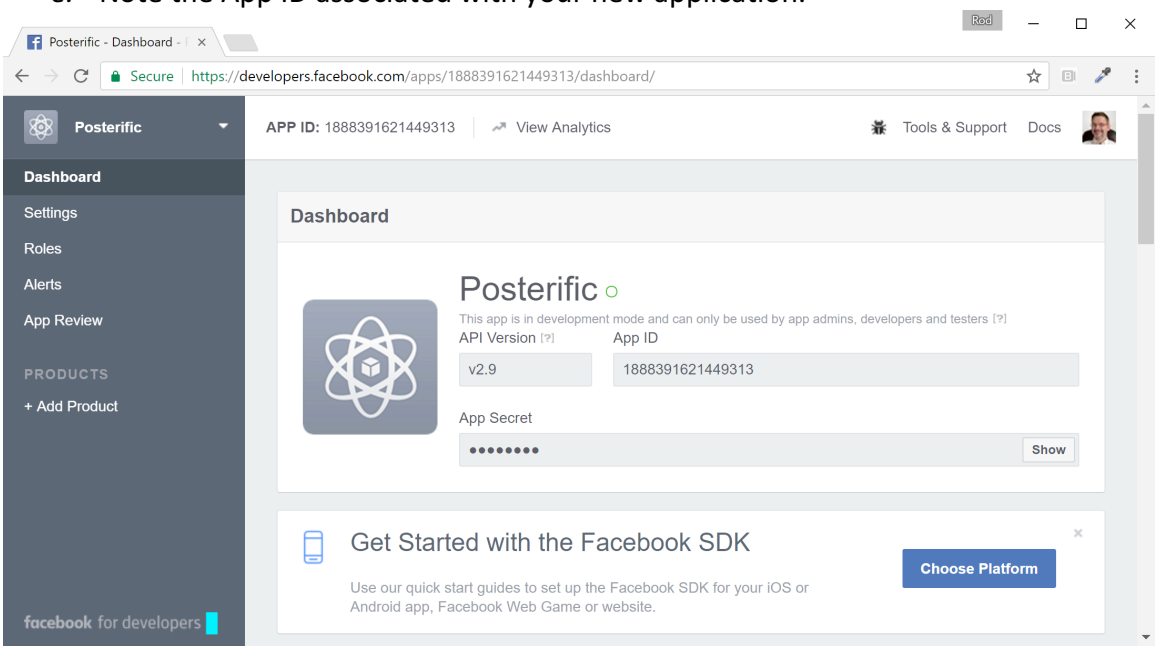
    c. Enter *Posterific* as the Display Name and put your own email address in the Contact Email field. Click the Create App ID button.



    d. Enter the captcha and click the Submit button.

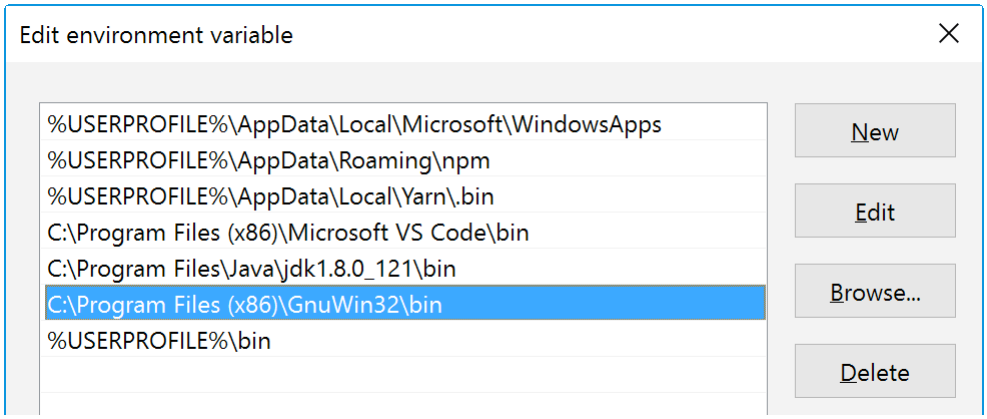


e. Note the App ID associated with your new application.



Install Open SSL

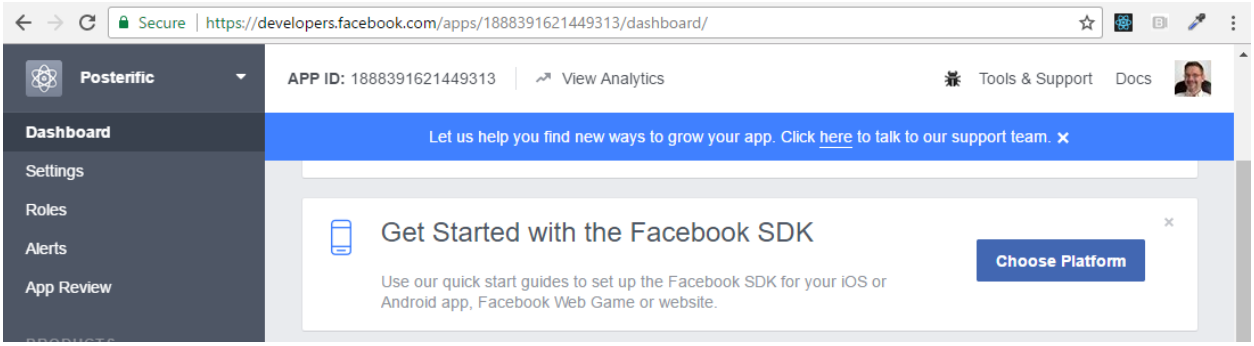
62. Download and run the Open SSL installer from https://sourceforge.net/projects/gnuwin32/files/openssl/0.9.8h-1/
63. Add C:\Program Files (x86)\GnuWin32\bin to your path environment variable.



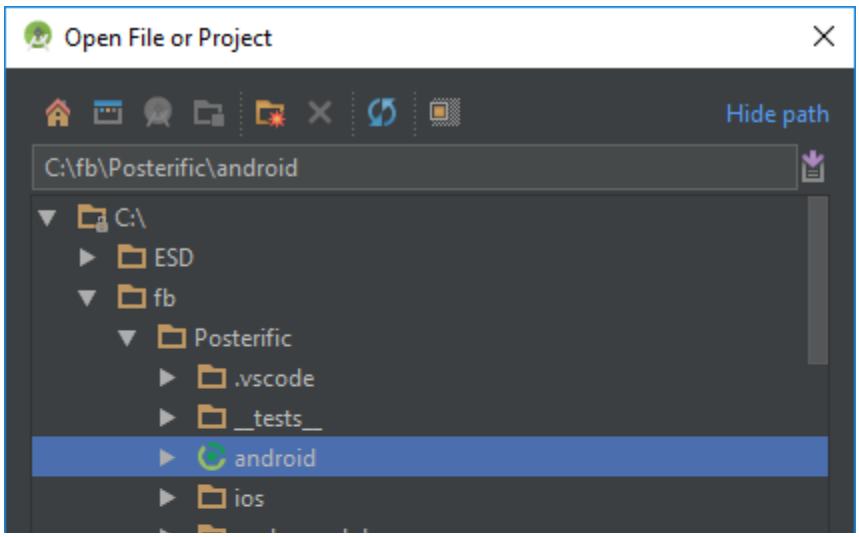
Note: openssl.exe is used to create a developer hash. You will complete this step as part of the process of integrating the Android platform into your Facebook application.

Configure your Facebook application for the Android platform

64. Return to the Application Dashboard in the browser and click the Choose Platform button. When the choices appear, select Android.



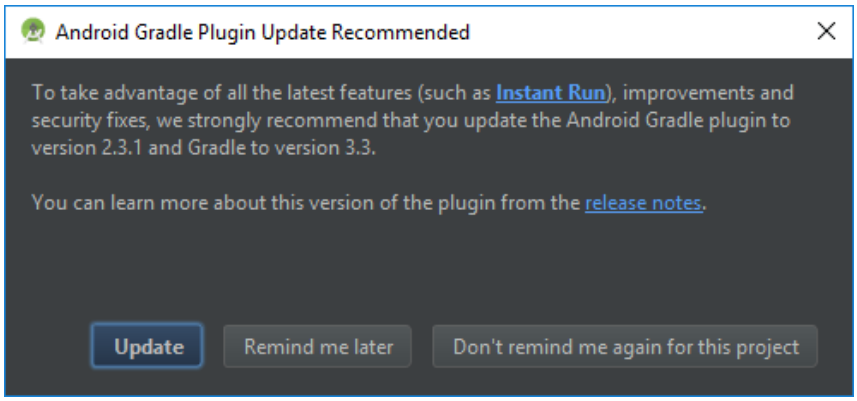
65. Open Android Studio and choose the *Open an existing Android Studio project* option from the welcome screen. Open the *C:\fb\Posterific\android* folder.



Note: You could make the edits described in this section in any editor but Android Studio will help you to detect and fix errors with greater ease.

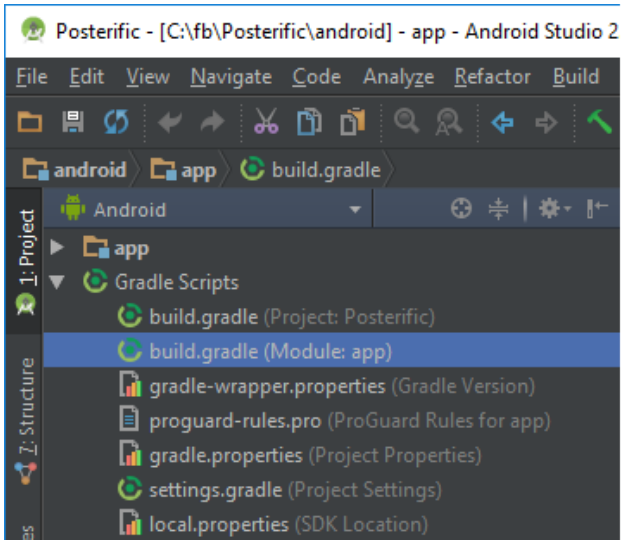
---

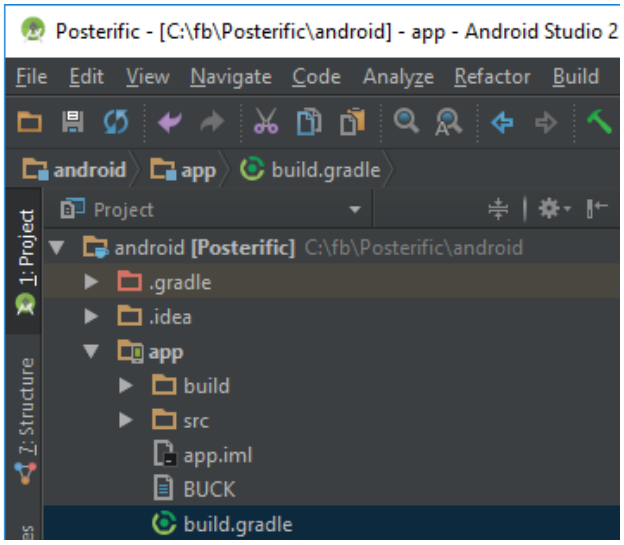
66. When you are prompted to upgrade the Gradle plugin, click the “Don’t remind me again for this project” button.



67. Open the module level build.gradle file.

Note: The project contains two files called build.gradle. See the illustrations below for help selecting the right one.





68. Add the facebook sdk to the dependencies block near the bottom of the file. Add the last compile line as shown below:

```
dependencies {
    compile fileTree(dir: "libs", include: ["*.jar"])
    compile "com.android.support:appcompat-v7:23.0.1"
    compile "com.facebook.react:react-native:+"  // From node_modules
    compile 'com.facebook.android:facebook-android-sdk:4.22.1'
}
```

69. Select Build > Make Project

Configure your Facebook App ID

70. Return to your App Dashboard and copy the facebook_app_id string.

```
<string name="facebook_app_id">1277634958939727</string>
```

Note: Each application has a unique ID so the value will differ from application to application.

---

71. Open */app/src/main/res/values/strings.xml*.
72. Paste the <string…> node you copied from the dashboard.
73. That file should now look like this:

```
<resources>
    <string name="app_name">Posterific</string>
    <string name="facebook_app_id">xxxxxxxxxxxxxxxx</string>
</resources>
```

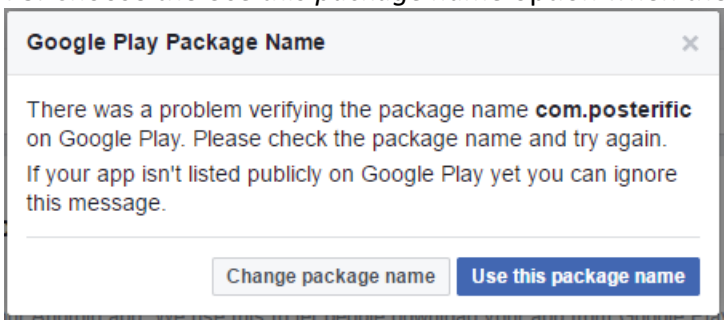
where xxxxxxxxxxxxxxx corresponds to your own application ID.

74. Open */app/src/main/res/AndroidManifest.xml*.
75. Add the metadata tag to the <application…> node:

```
<meta-data
    android:name="com.facebook.sdk.ApplicationId"
    android:value="@string/facebook_app_id"/>
```

Configure Android project

76. In the Tell us about your Android project section,
    a. enter the Package Name: *com.posterific*
    b. enter the Default Activity Class Name: *com.posterific.MainActivity*
77. Click the Next button.
78. Choose the *Use this package name* option when the warning appears.



Configure your development key hash.

79. In the Key Hashes section, click the link to Show how to generate a development key hash.

Note: The release key hash is not required at this time.

---

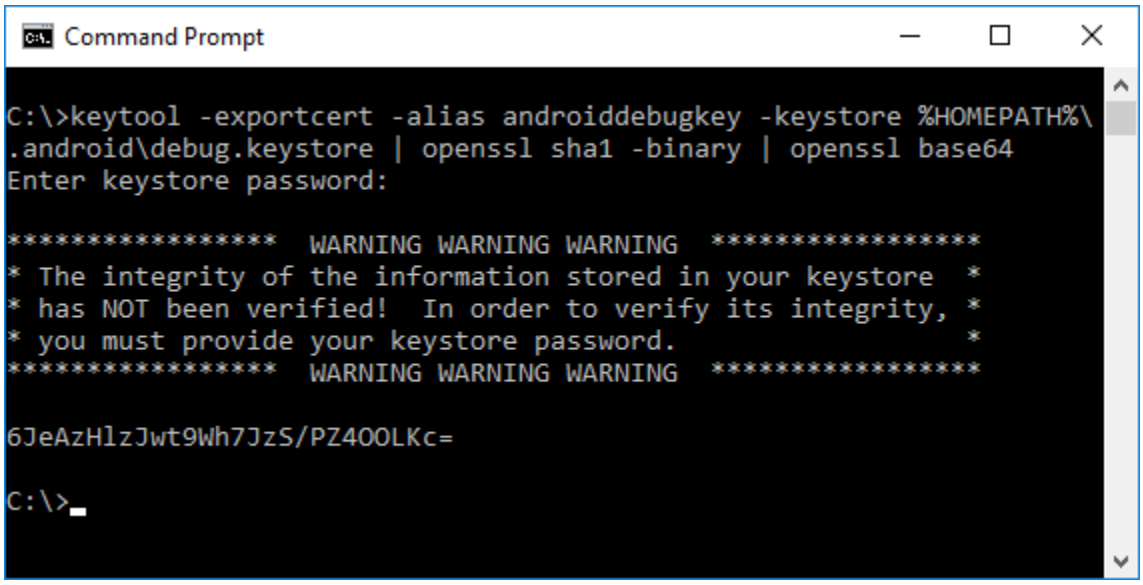
80. Copy the command listed in the Windows section and paste it into a command prompt.

```
keytool -exportcert -alias androiddebugkey -keystore
%HOMEPATH%\.android\debug.keystore | openssl sha1 -binary | openssl base64
```

Note: This command is listed in *{student-files}\commands.txt*.

---

81. Leave the password prompt blank.



82. Copy the 28-character code into the key hashes form in the app dashboard and click the Next button.

Note: The final two sections (*Track App Installs and App Opens* and *Next Steps*) apply specifically applications build directly with the Android SDK so you can skip those sections for now.

Congratulations! You are now ready to start building Facebook Platform features into your application.