

# React Native Setup & Debugging

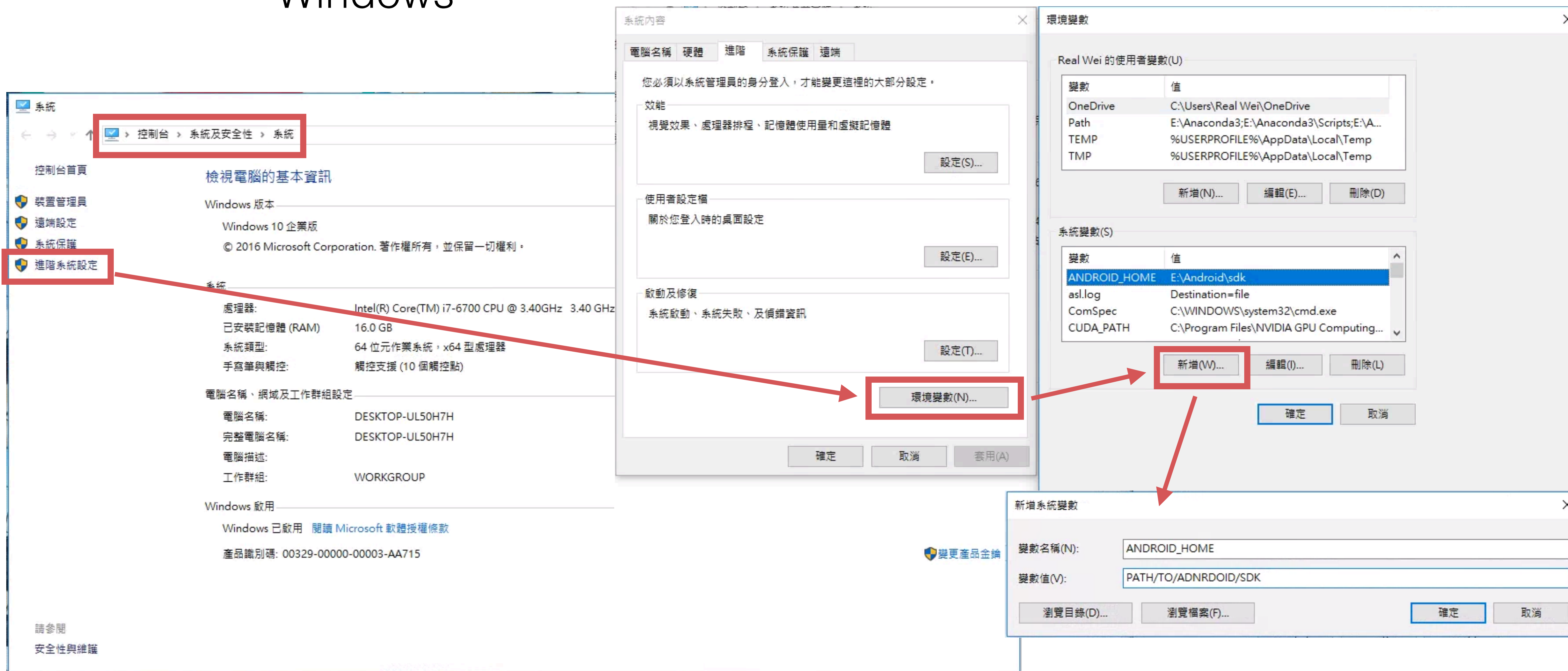
DataLab

# Installation

- Requirements
  - [JDK 8](#)
  - [Python](#) : both 2 and 3 works for react native
  - [Android Studio](#)
  - React Native : `npm install -g react-native-cli`
- To run on iOS, you will need Xcode installed on Mac
  - You can setup a Mac VM

# Environment Variable

- Windows

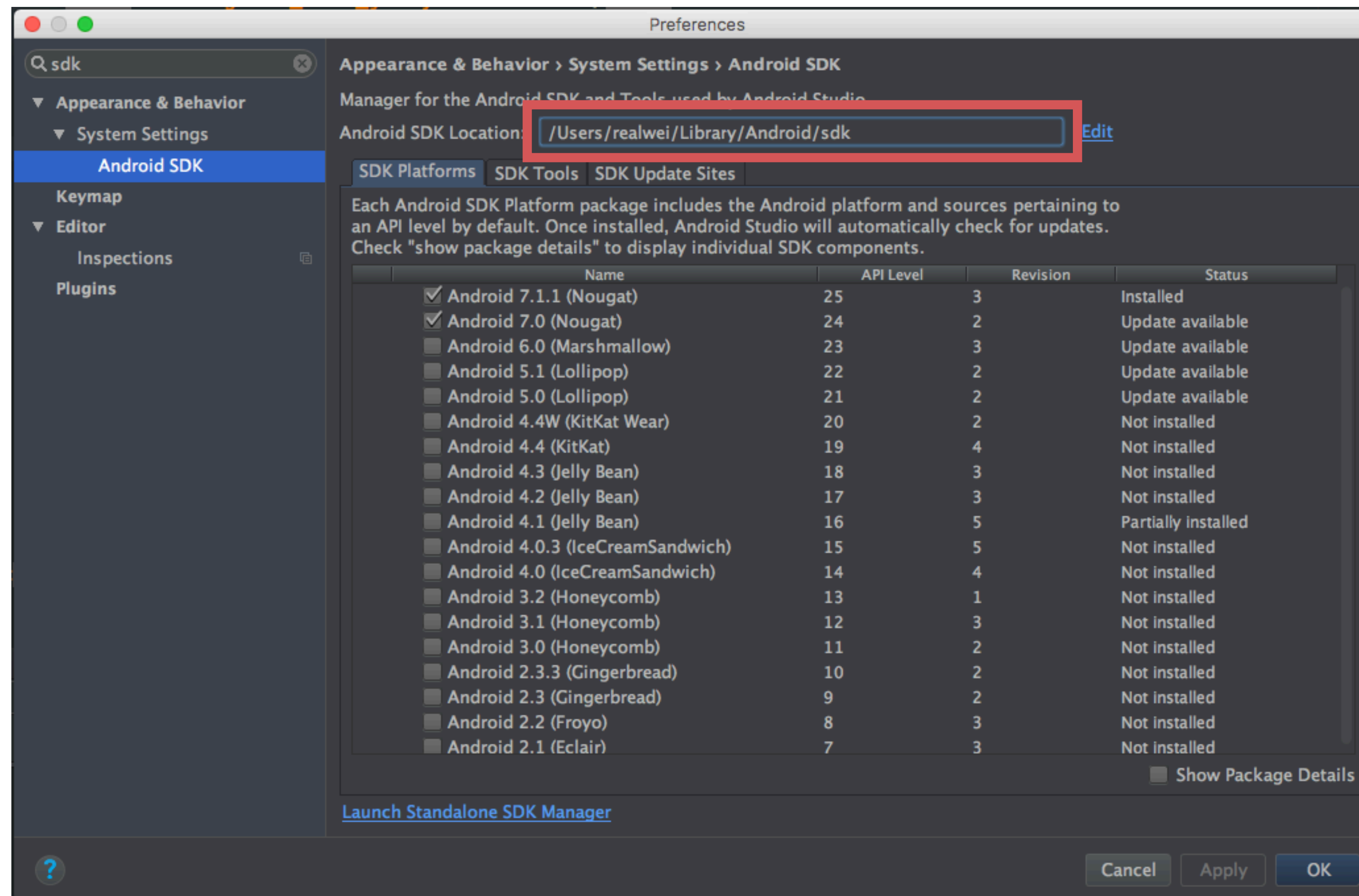


# Environment Variable

- Mac
  - In terminal  
`vim ~/.bash_profile` (or `~/.profile`)
  - Add following lines  
`export ANDROID_HOME='path/to/androidSDK/'` (see next slide)  
`export PATH=$ANDROID_HOME/tools:$PATH`  
`export PATH=$ANDROID_HOME/platform-tools:$PATH`
  - In terminal  
`source ~/.bash_profile` (or `~/.profile`)

# Android SDK Path

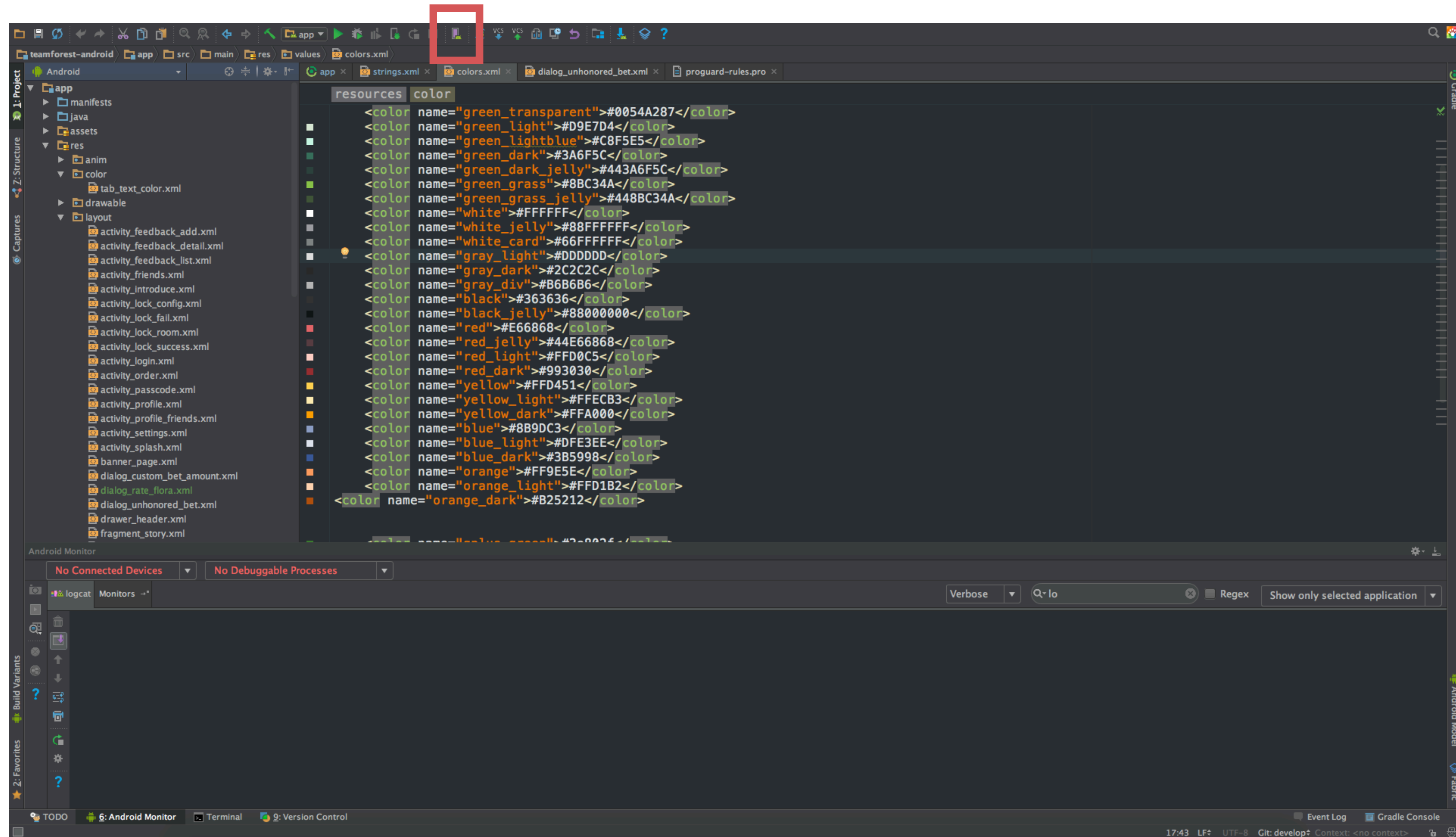
- Preferences



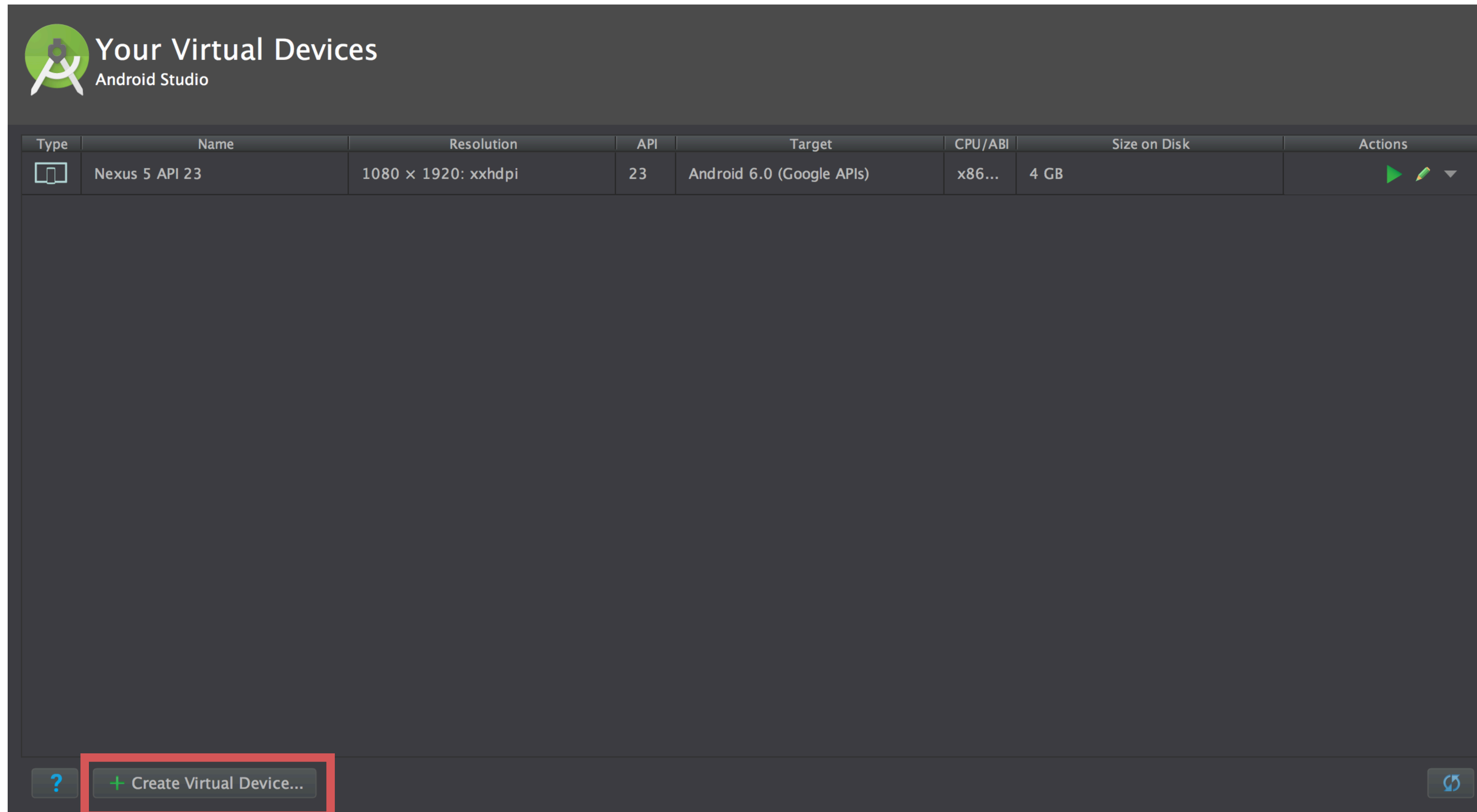


# Android Virtual Device (AVD) Manager

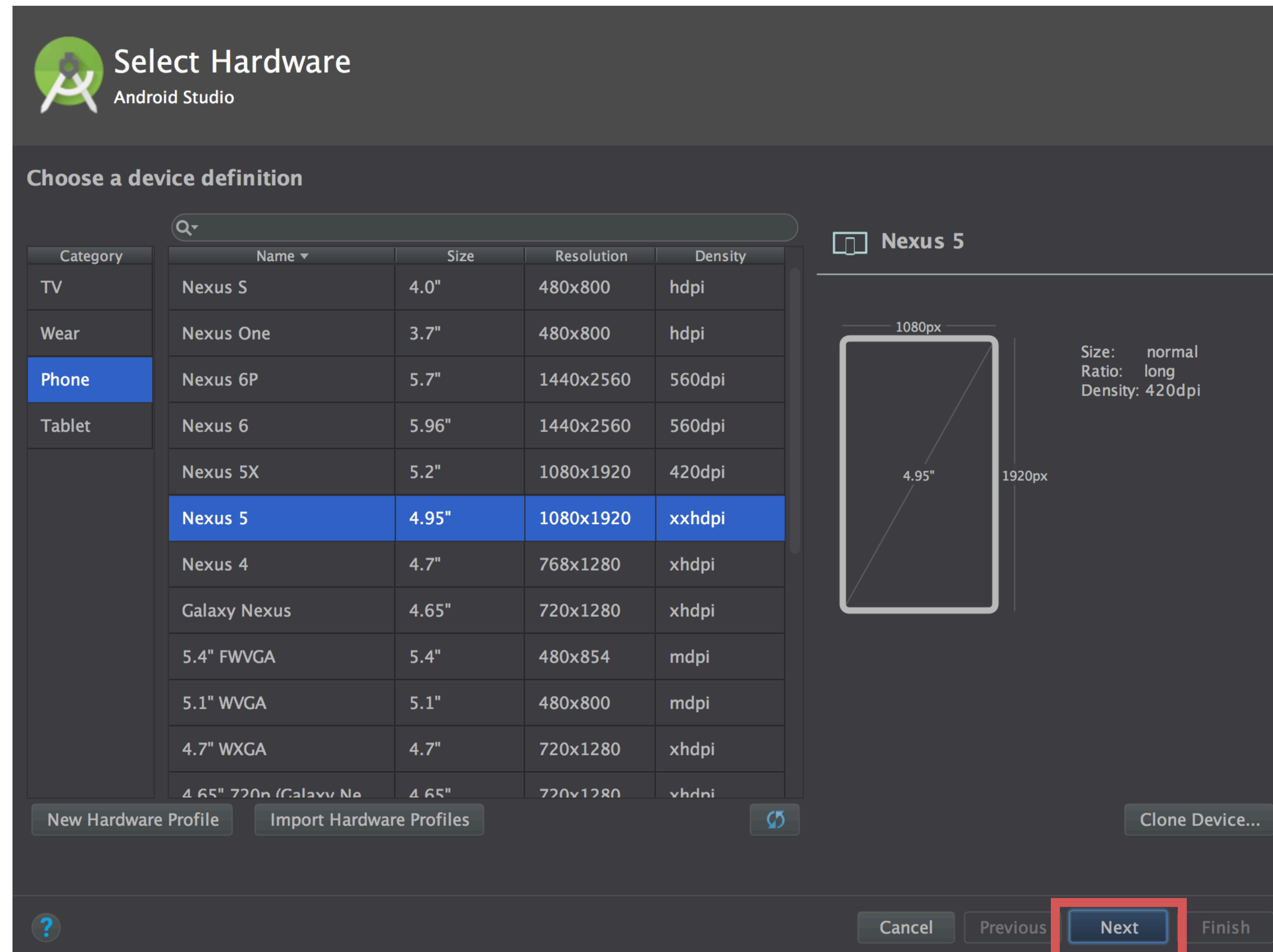
- Launch Android Studio



# Create Virtual Device



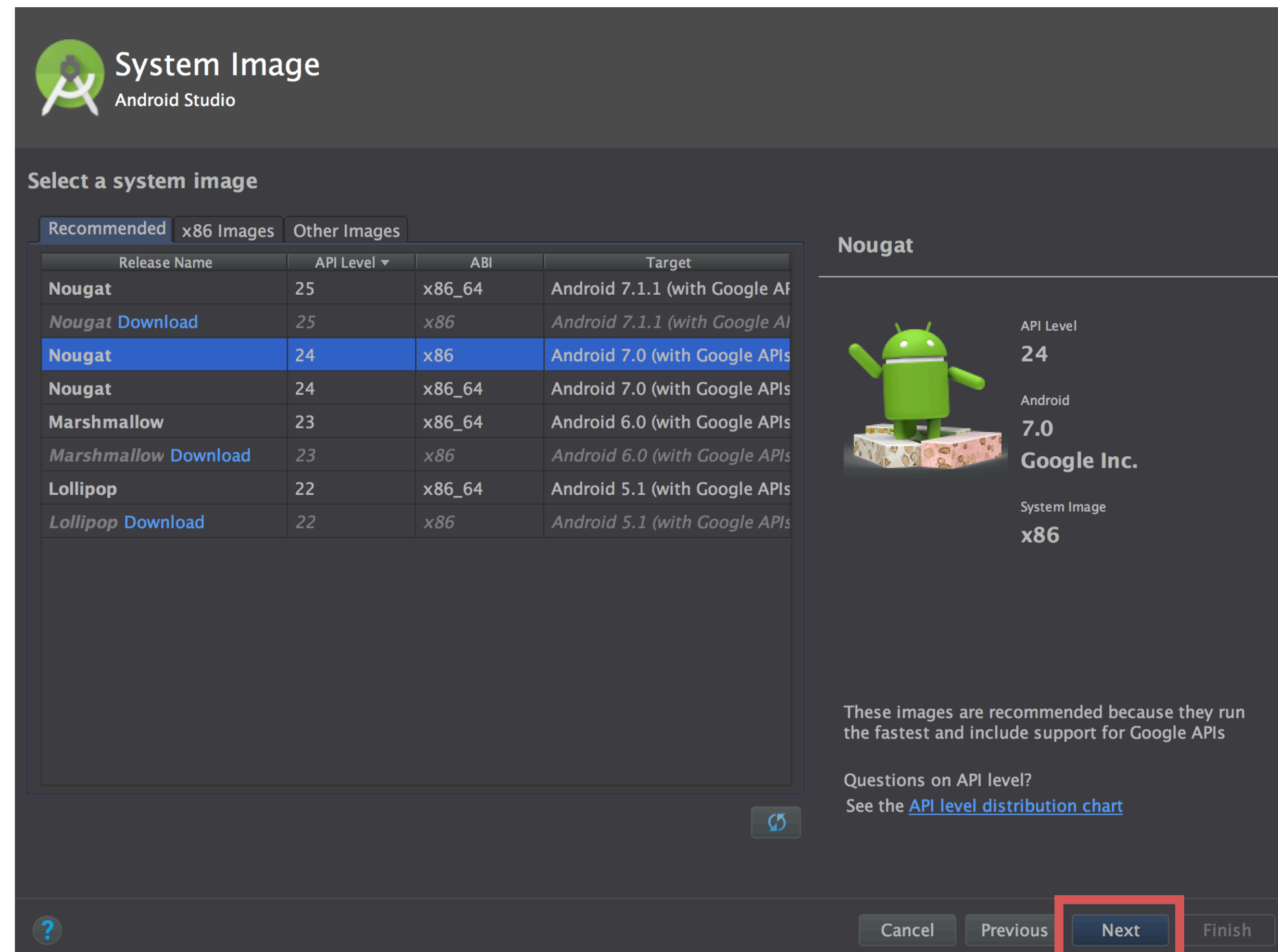
# Create Virtual Device






# Create Virtual Device

- Choose and download any system image you like




# Create Virtual Device

**Android Virtual Device (AVD)**  
Android Studio


**Verify Configuration**

AVD Name

 Nexus 5

4.95 1080x1920 420dpi

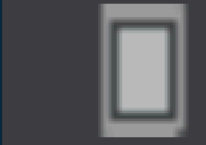
Change...


 Nougat

Android 7.0 x86

Change...

Startup orientation

 Portrait

 Landscape

Emulated Performance

Graphics:

Device Frame

☐ Enable Device Frame

Show Advanced Settings

**Enable device frame**  
Enable a frame around the Android emulator window that mimics the look of a real Android device. Click on Show Advanced Settings for more options.

?

Cancel

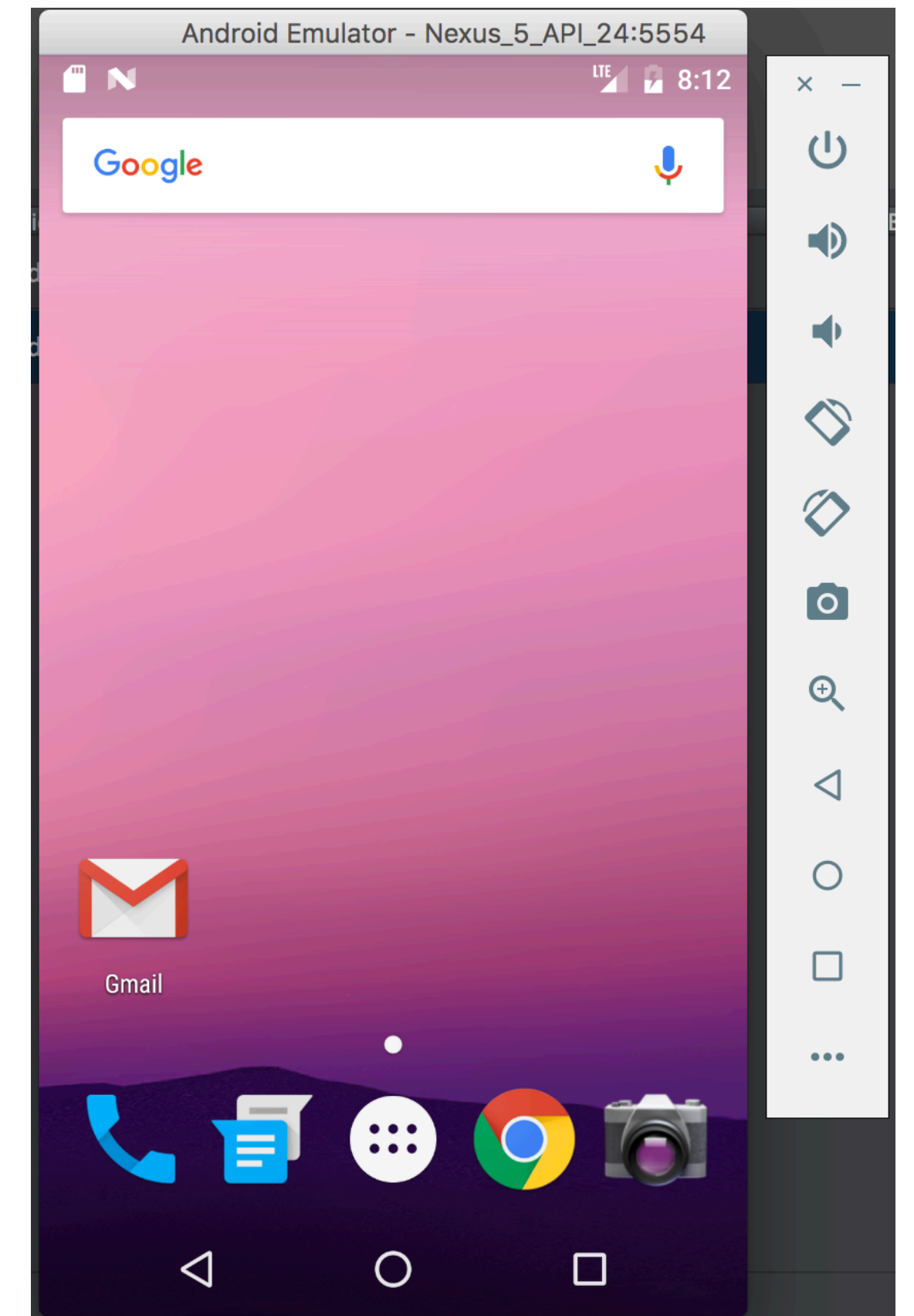
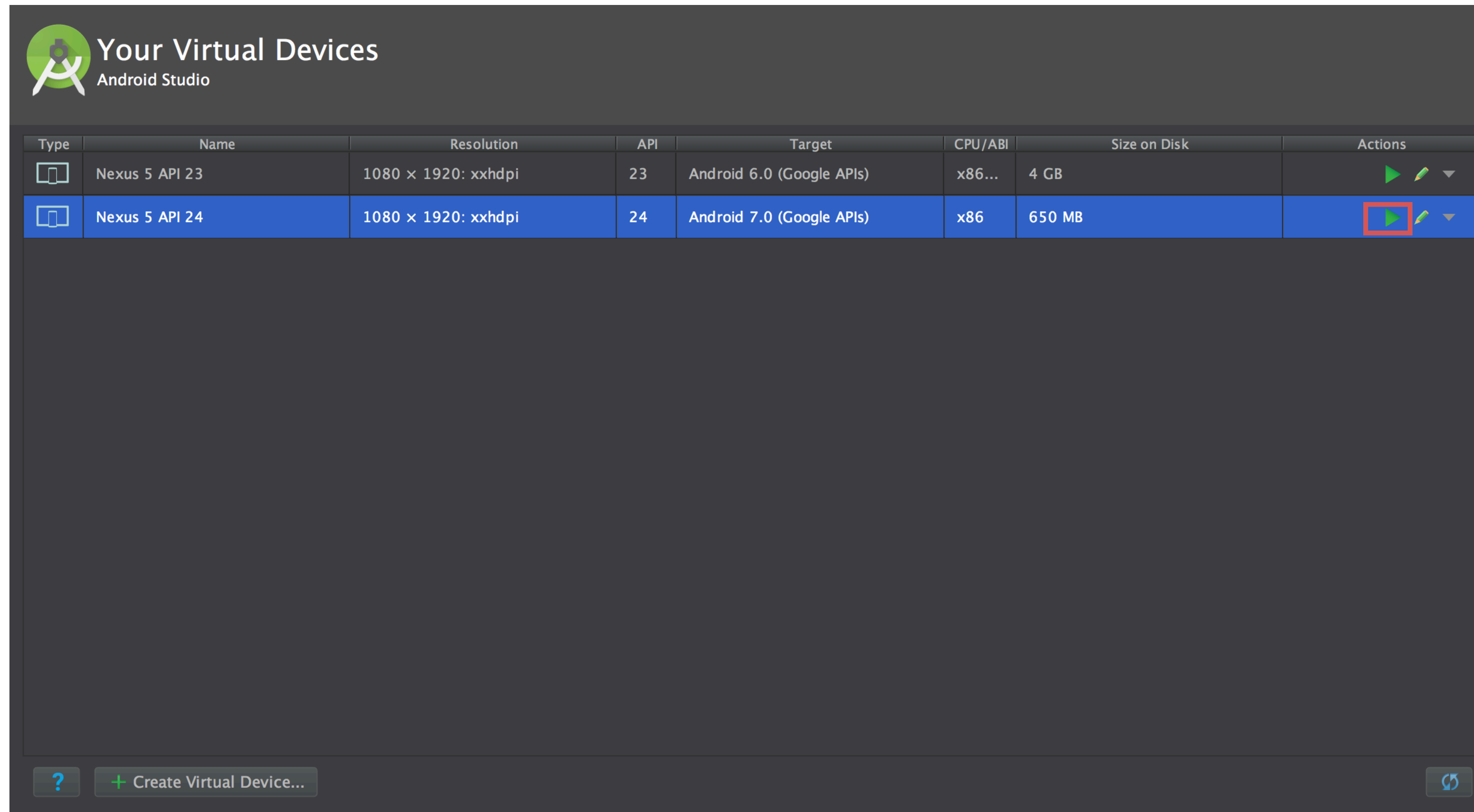
Previous

Next

Finish

Disable Device Frame

# Create Virtual Device



# Create a RN Project

```
~/Documents/Projects/ss ➔ react-native init ReactTimer
This will walk you through creating a new React Native project in /Users/
Installing react-native...
Consider installing yarn to make this faster: https://yarnpkg.com
ReactTimer@0.0.1 /Users/realwei/Documents/Projects/ss/ReactTimer
├─ UNMET PEER DEPENDENCY react@16.0.0-alpha.6
├─ react-native@0.44.0
├─ absolute-path@0.0.0
├─ art@0.10.1
├─ async@2.4.1
├─ babel-core@6.24.1
├─ babel-code-frame@6.22.0
├─ js-tokens@3.0.1
├─ babel-helpers@6.24.1
├─ babel-messages@6.23.0
├─ babel-template@6.24.1
├─ convert-source-map@1.5.0
├─ json5@0.5.1
```

⋮

## To run your app on iOS:

```
cd /Users/realwei/Documents/Projects/ss/ReactTimer
react-native run-ios
- or -
Open ios/ReactTimer.xcodeproj in Xcode
Hit the Run button
```

## To run your app on Android:

```
cd /Users/realwei/Documents/Projects/ss/ReactTimer
Have an Android emulator running (quickest way to get started) or a device connected
react-native run-android
```

```
~/Documents/Projects/ss ➔
```

# Run a RN Project

```
~/Documents/Projects/ss/ReactTimer react-native run-android
Scanning 555 folders for symlinks in /Users/realwei/Documents/Projects/ss/ReactTimer
JS server already running.
Building and installing the app on the device (cd android && ./gradlew installDebug)
Incremental java compilation is an incubating feature.
:app:preBuild UP-TO-DATE
:app:preDebugBuild UP-TO-DATE
:app:checkDebugManifest
:app:preReleaseBuild UP-TO-DATE
:app:prepareComAndroidSupportAppcompatV7220Library
```

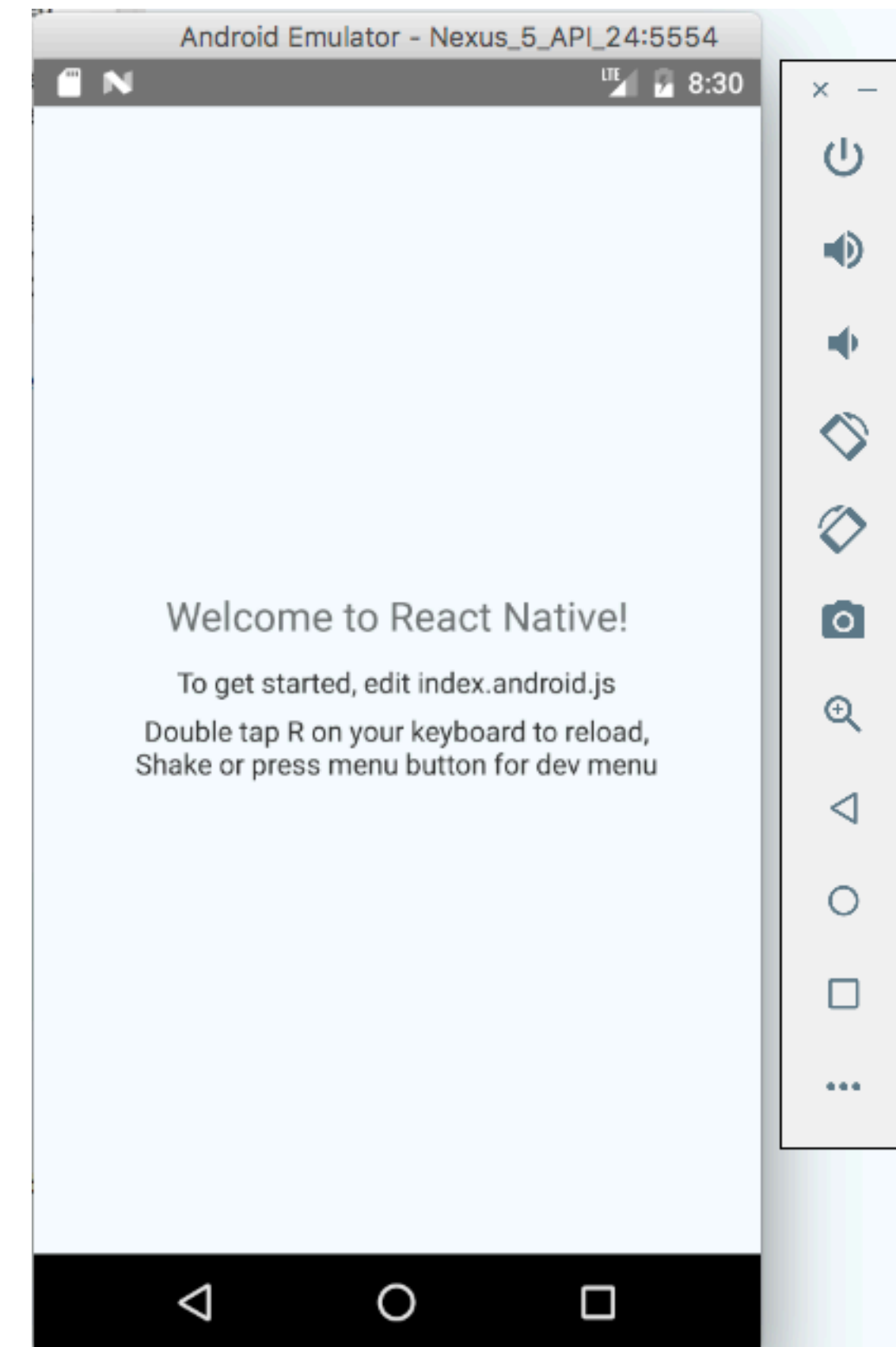
⋮

```
:app:validateSigningDebug
:app:packageDebug
:app:assembleDebug
:app:installDebug
Installing APK 'app-debug.apk' on 'Nexus_5_API_24(AVD) - 7.0' for app:debug
Installed on 1 device.

BUILD SUCCESSFUL

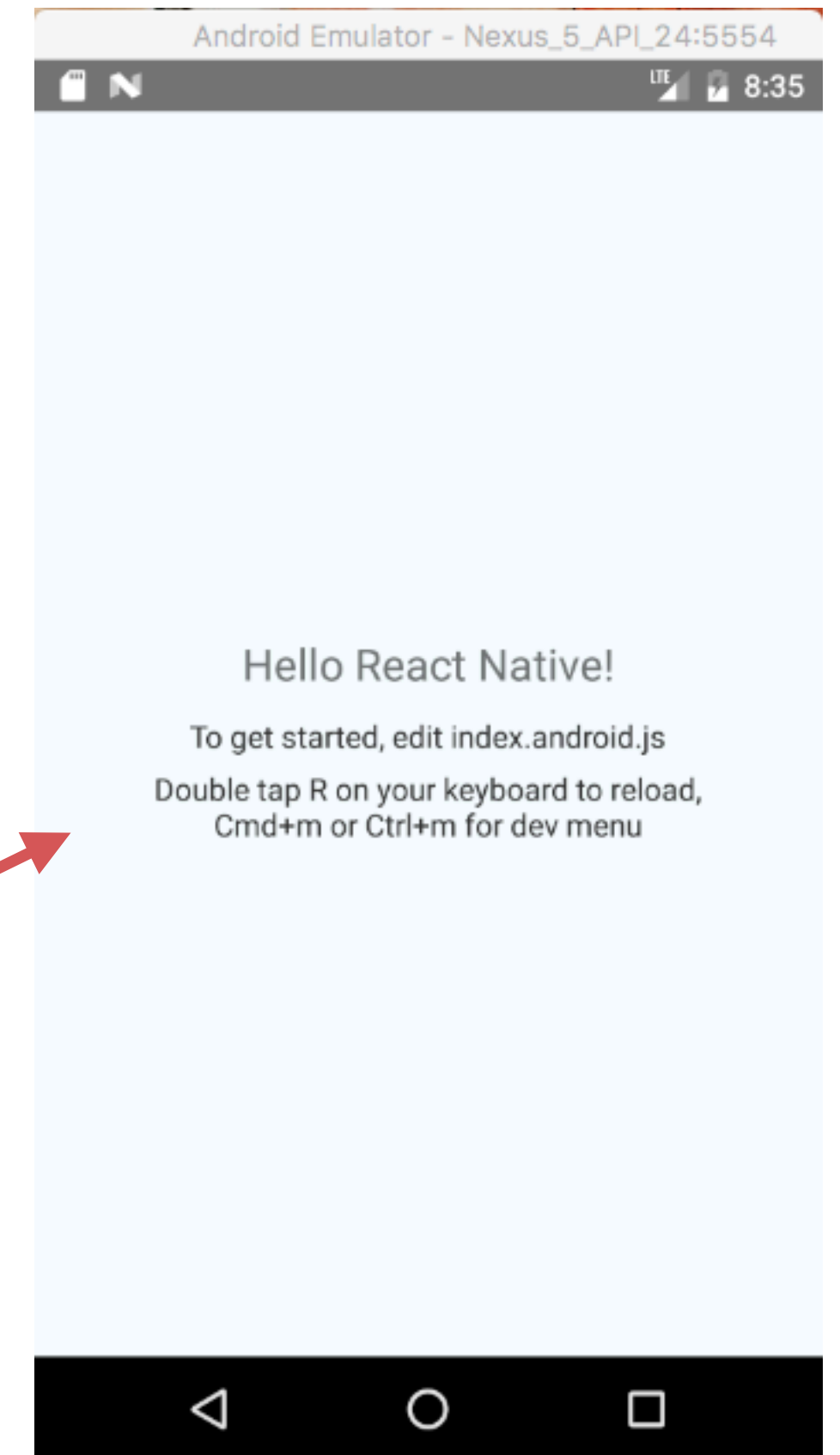
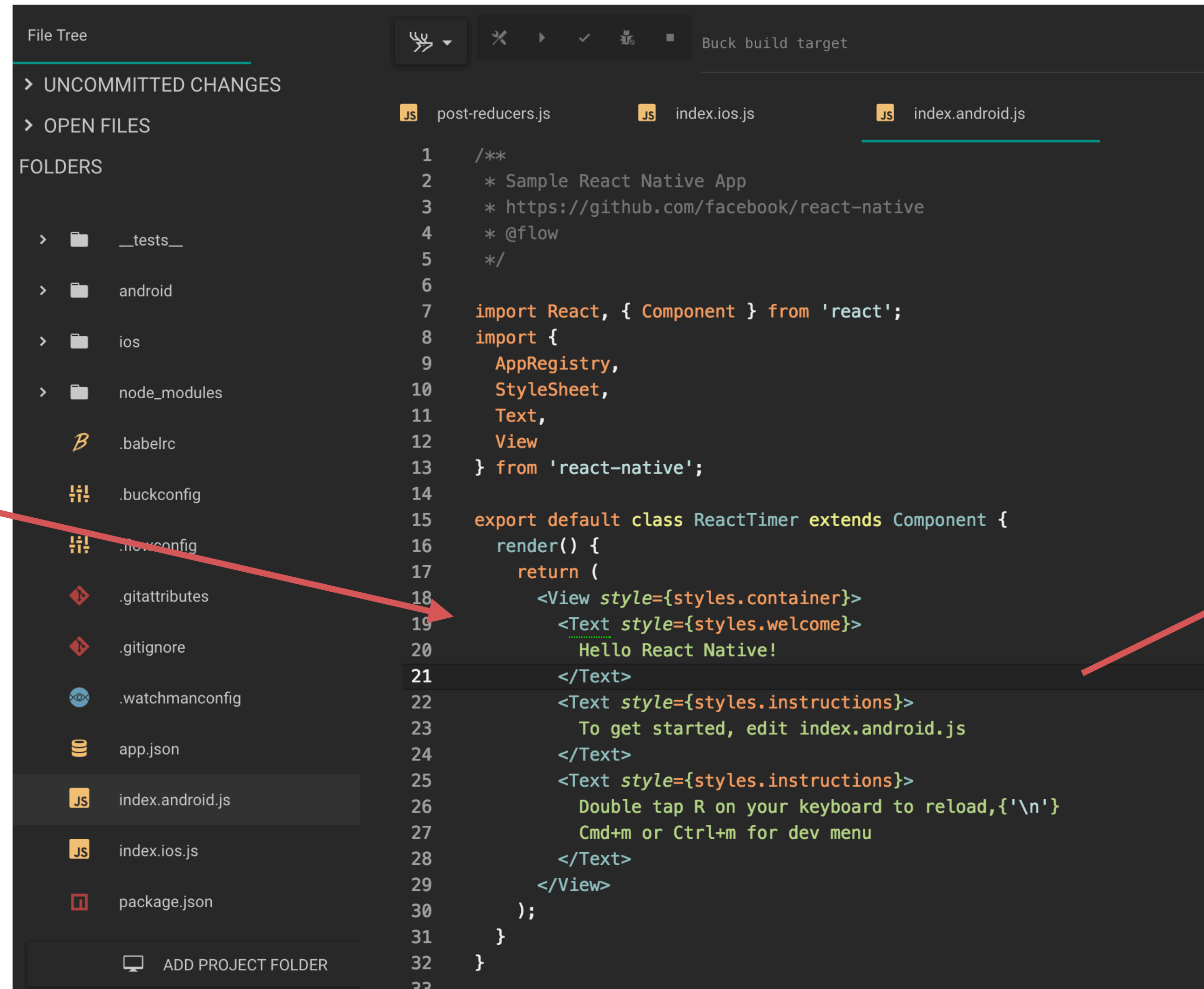
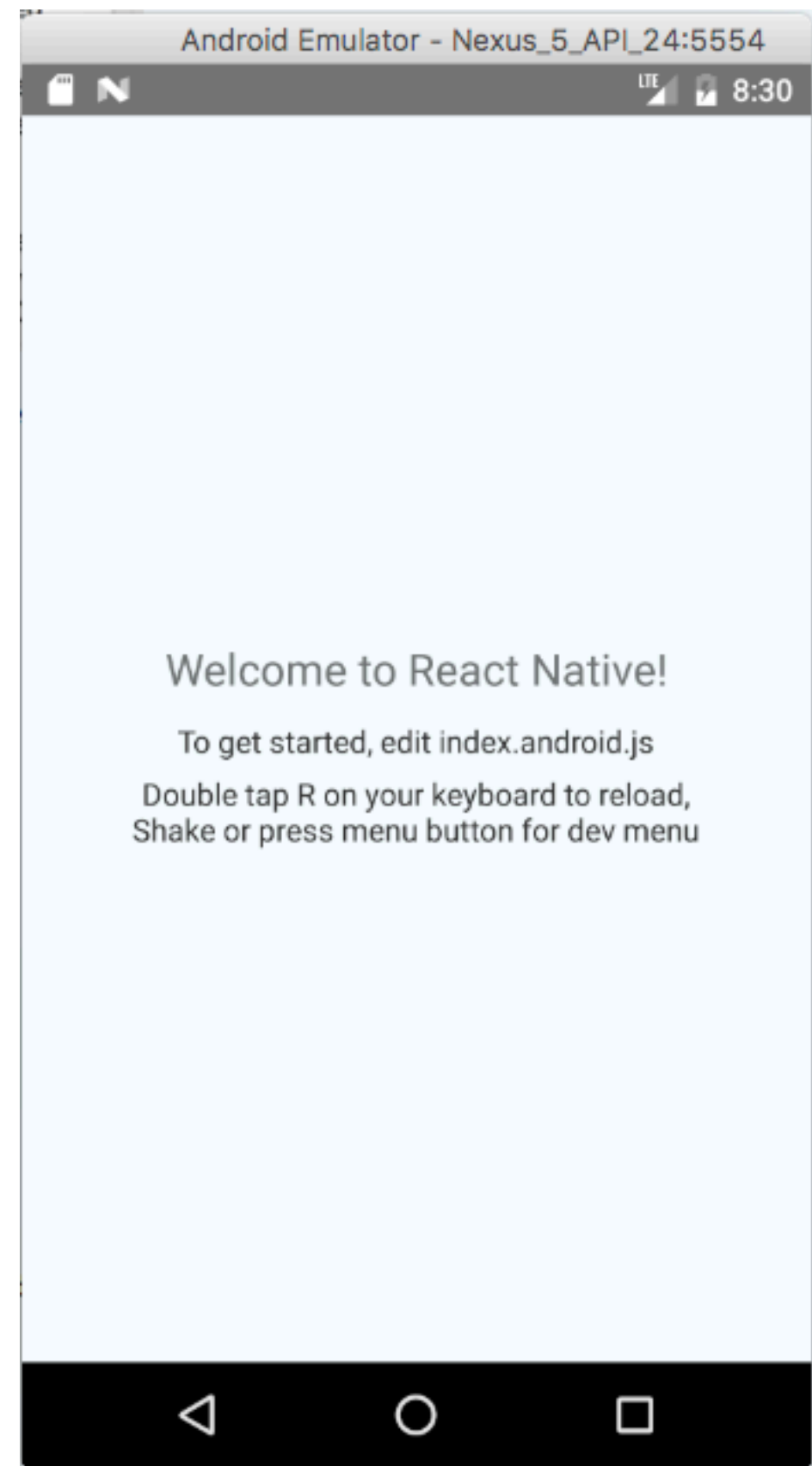
Total time: 1 mins 16.489 secs

This build could be faster, please consider using the Gradle Daemon: https://docs.gradle.org/4.10.1/userguide/gradle_daemon.html
Running /Users/realwei/Library/Android/sdk/platform-tools/adb -s emulator-5554 shell
Starting the app on emulator-5554 (/Users/realwei/Library/Android/sdk/platform-tools/adb -s emulator-5554 shell
acttimer/.MainActivity)...
Starting: Intent { cmp=com.reacttimer/.MainActivity }
~/Documents/Projects/ss/ReactTimer
```

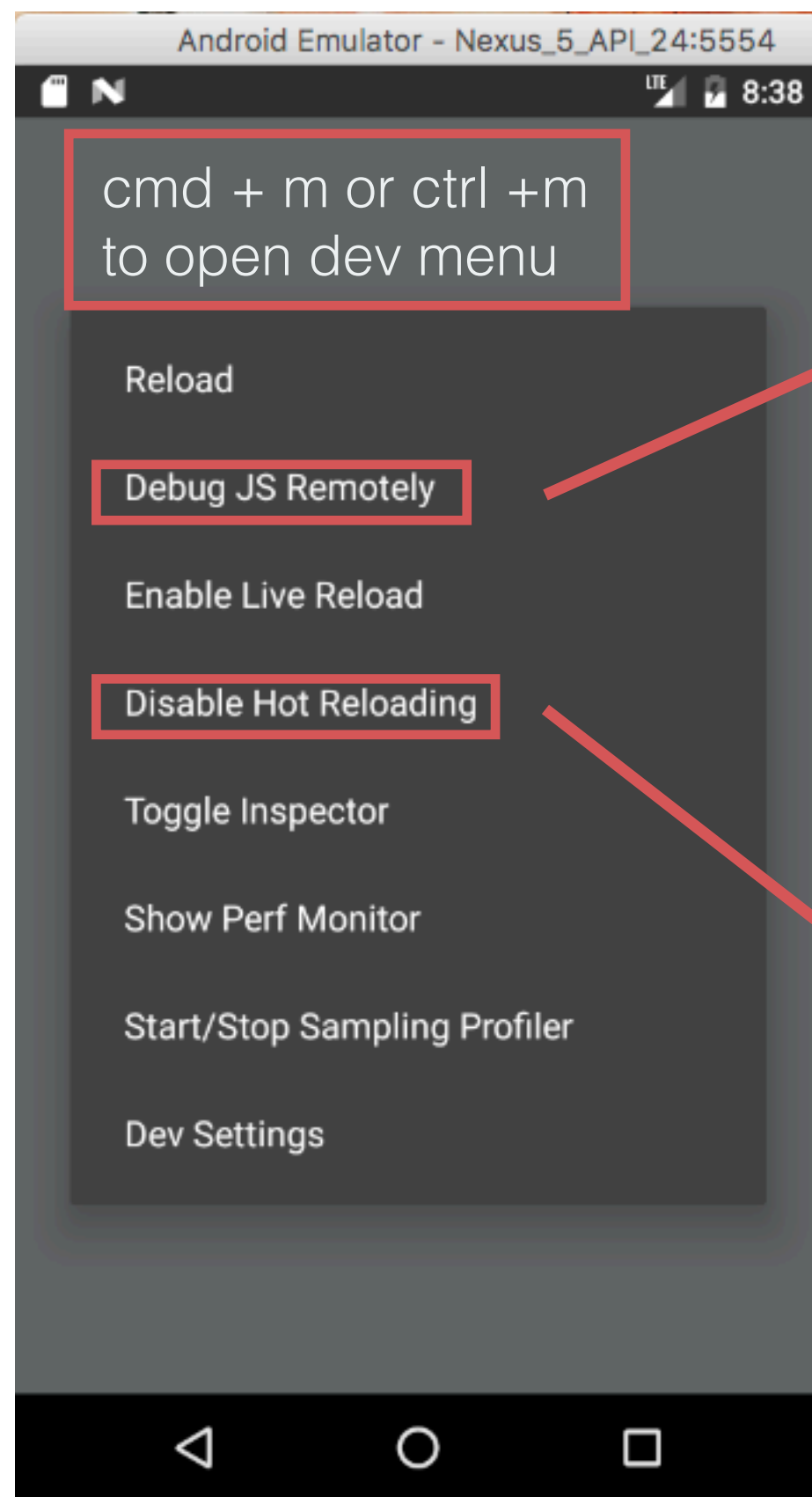




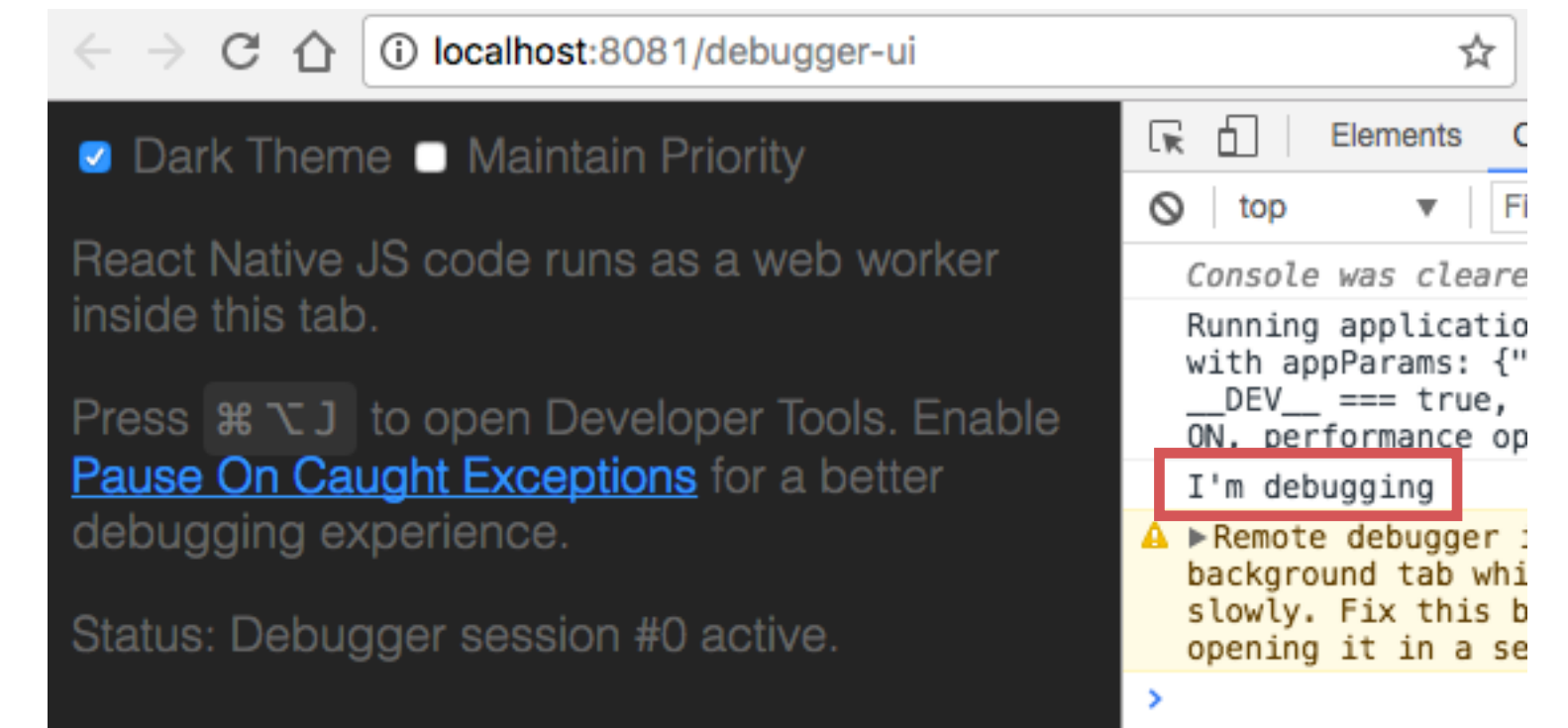
# Modify a RN Project



# Debug a RN Project



```
export default class ReactTimer extends Component {  
  render() {  
    console.log('I\'m debugging');  
    return (  
      <View style={styles.container}>  
        <Text style={styles.welcome}>  
          Hello React Native!  
        </Text>  
        <Text style={styles.instructions}>  
          To get started, edit index.android.js  
        </Text>  
        <Text style={styles.instructions}>  
          Double tap R on your keyboard to reload,{'\n'}  
          Cmd+m or Ctrl+m for dev menu  
        </Text>  
      </View>  
    );  
  }  
}
```



Enable hot reloading will reload your app whenever you save your modification (just like webpack watch)

# Today's Mission

- Countdown 5 seconds in the virtual device.
  - Start countdown when the button is pressed.
  - The timer should stop at 0, or when the component is unmounted.
  - Find Button usage in the [React Native Docs](#)
- Hint
  - use `setInterval()`
  - call `clearInterval()` in `componentWillUnmount()`
  - `setInterval()` will return timer id, store it and pass it to `clearInterval()`
- Deadline : **5/23 23:59**