**REACT NATIVE INSTALL UBUNTU 18.04 VIRTUAL BOX**

<https://reactnative.dev/docs/environment-setup>

sudo snap install node --classic --channel=12

npx -v

6.13.4

npm -v

6.13.4

node -v

v12.15.0

sudo apt-get install openjdk-8-jre

sudo apt-get install openjdk-8-jdk

Android Studio through Ubuntu Software

OR

<https://developer.android.com/studio>

https://developer.android.com/studio/install#linux

sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 libbz2-1.0:i386

tar -zxvf android-studio-ide-192.6308749-linux.tar.gz

cd ~/android-studio/bin/studio.sh

Choose a "Custom" setup when prompted to select an installation type. Make sure the boxes next to all of the following are checked:

Android SDK

Android SDK Platform

Android Virtual Device

Building a React Native app with native code, however, requires the Android 9 (Pie) SDK in particular. Additional Android SDKs can be installed through the SDK Manager in Android Studio.

The SDK Manager can be accessed from the "Welcome to Android Studio" screen. Click on "Configure", then select "SDK Manager".

Select the "SDK Platforms" tab from within the SDK Manager, then check the box next to "Show Package Details" in the bottom right corner. Look for and expand the Android 9 (Pie) entry, then make sure the following items are checked:

Android SDK Platform 28

Intel x86 Atom\_64 System Image or Google APIs Intel x86 Atom System Image

Next, select the "SDK Tools" tab and check the box next to "Show Package Details" here as well. Look for and expand the "Android SDK Build-Tools" entry, then make sure that 28.0.3 is selected.

Finally, click "Apply" to download and install the Android SDK and related build tools.

gedit $HOME/.bashrc

add these:

…

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

…

Then:

source $HOME/.bashrc

echo $PATH

sudo apt install curl

/bin/bash -c "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/master/install.sh>)"

sudo apt-get install build-essential

echo 'eval $(/home/linuxbrew/.linuxbrew/bin/brew shellenv)' >> /home/alberto/.profile

eval $(/home/linuxbrew/.linuxbrew/bin/brew shellenv)

brew install gcc

brew update

brew install watchman

**npx react-native init frontendMobile (ONLY CAMELCASE)**

Running your app on Android devices:

1. Enable Debugging over USB

Most Android devices can only install and run apps downloaded from Google Play, by default. You will need to enable USB Debugging on your device in order to install your app during development. To enable USB debugging on your device, you will first need to enable the "Developer options" menu by going to Settings → About phone and then tapping the Build number row at the bottom seven times. You can then go back to Settings → System → Developer options to enable "USB debugging".

2. Plug in your device via USB

Let's now set up an Android device to run our React Native projects. Go ahead and plug in your device via USB to your development machine.

Next, check the manufacturer code by using lsusb (on mac, you must first install lsusb). lsusb should output something like this:

$ lsusb

Bus 001 Device 003: ID 22b8:2e76 Motorola PCS

From the above line, you want to grab the first four digits from the device ID:

22b8:2e76

In this case, it's 22b8. That's the identifier for Motorola.

You'll need to input this into your udev rules in order to get up and running:

echo 'SUBSYSTEM=="usb", ATTR{idVendor}=="22b8", MODE="0666", GROUP="plugdev"' | sudo tee /etc/udev/rules.d/51-android-usb.rules

Make sure that you replace 22b8 with the identifier you get in the above command.

Now check that your device is properly connecting to ADB, the Android Debug Bridge, by running adb devices.

sudo apt install adb

$ adb devices

List of devices attached

emulator-5554 offline # Google emulator

14ed2fcc device # Physical device

You must have only one device connected at a time.

3. Run your app

Type the following in your command prompt to install and launch your app on the device:

Install app:

one terminal:

cd ~/Desktop/AWM-CHESS/AVM-CHESS/frontendMobile/frontendMobile

npm start

another terminal:

cd ~/Desktop/AWM-CHESS/AVM-CHESS/frontendMobile/frontendMobile

npx react-native run-android

App already installed:

one terminal:

npm start

then open application in the phone

**SCREEN PHONE IN UBUNTU**:

$ which adb

/usr/bin/adb

<https://github.com/Genymobile/scrcpy> download from Ubuntu Software

$ ADB="/home/alberto/Android/Sdk/platform-tools/adb" scrcpy

<https://github.com/Genymobile/scrcpy/issues/541>

PROBLEMS:

# If you are sure the module exists, try these steps:

1. Clear watchman watches: watchman watch-del-all

2. Delete node\_modules: rm -rf node\_modules && rm package-lock.json && npm install --save

3. Reset Metro's cache: rm -rf /tmp/metro-\*

4. Remove the cache: npx react-native start --reset-cache

5. Rebuild: npx react-native run-android