Using iOS and Android Simulators
published 1-23-2020

These instructions are for students who wish to use the XCode or Android Studio simulators instead of a physical device with Expo.

keep in mind, emulators require a lot of RAM, so you may find that bundling and displaying code takes much longer than when using a physical device.

The official Expo docs with some troubleshooting tips are available here:

https://docs.expo.io/versions/latest/workflow/ios-simulator/
https://docs.expo.io/versions/latest/workflow/android-studio-emulator/

macOS

XCode iOS Simulator

1. Download the full version of Xcode from the App Store. This is a huge file at around 7GB, so it can take a LONG time to download and install. It might seem like it it is stuck and not installing, but it is probably not.

2. Launch XCode and agree to its terms. It will start installing more tools and software

3. Eventually, you will get a "Welcome to XCode" screen

4. In the top menu bar, click "XCode", then "Preferences" and then "Locations".

5. Make sure that the "Command Line Tools" are installed and selected.

6. In the top menu bar, click XCode, then "Open Developer Tool" then click "Simulator"

7. This will open an iPhone simulator on your machine. If everything is working well, it should load a generic screen with a few apps.

8. Go to your terminal and in your React Native project directory, run `npm start`
9. When the Metro Bundler opens up in your browser, click on the option in the left sidebar to “Run on iOS simulator”

10. If you navigate back to your iPhone simulator you should see an “Open In Expo” button. Click Open.

11. This will launch the Expo application in the simulated device and run your bundled code. Eventually, it should load the “Hi There!” screen.

Android Studio Simulator

1. Install Android Studio: https://developer.android.com/studio
2. After downloading, run the installer.

3. A dialog box should pop up. Do not Import Settings, Just Click "Ok"

4. Click “Next" to go thru the Wizard

5. Click “Next” to do a Standard install

6. Choose light or dark theme for your editor and click “Next”

7. Click “Finish”

8. You will likely get a dialog box pop and say “HAXM Installation wants to make changes”. Enter your password and click OK

9. If you get a “System Extension Blocked” error, follow the prompt to Open Security Preferences and click “Allow” for “Intel Corporation Apps”

10. Once the HAXM installer finished you should get a dialog box from Android Studio.

11. Click “AVD Manager”

12. Click “Create New Virtual Device”

13. Select a Device from the list and click “Next”

14. If the System Image for the most recent SDK is not installed, you will need to click the "Download" link next to the SDK version.

15. After the image has been downloaded, click “Next”

16. Preview the settings and then click “Finish”

17. You should see the newly created device in the “Your Virtual Device” screen.

18. Click the Play ▶ button all the way to the right in the "Actions" column.

19. This will open an Android simulator on your machine. If everything is working well, it should load a generic screen with a few apps.

20. Go to your terminal and in your React Native project directory, run `npm start`
21. When the Metro Bundler opens up in your browser, click on the option to “Run on Android simulator”

22. If you navigate back to your Android simulator you might see a “Permit Drawing Over Other Apps” message. Click the OK button and it should take you to the system settings screen to enable this. You will likely have to go back to the Metro Bundler and select “Run on Android simulator” again.

23. This will launch the Expo application in the simulated device and run your bundled code. Eventually, it should load the “Hi There!” screen.

Windows

ios Simulator

Unfortunately, you cannot run iOS simulators on Windows. You can, however, use a physical iOS device running Expo with your Windows machine.

Android Studio Simulator

Make sure Virtualization is enabled in your BIOS. Also, note, your processor must support HAXM. If it does not (which is common with many AMD or non-Intel machines) you will likely not be able to run the simulators. HAXM will also conflict with the Hyper-V Manager if it is enabled.

1. Install Android Studio: https://developer.android.com/studio
2. After Downloading run the installer.

3. Click “Next" to go through the installer wizard.

4. Click “Next” on Choose Components screen

5. Click “Next” on the Configuration Settings screen

6. Click “Install”

7. After completion click “Next”

8. Click “Finish”

9. In the next dialog box that pops up, do not Import settings, Just Click "Ok"

10. Click “Next" to go thru the Wizard

11. Click “Next” to do a Standard install

12. Choose light or dark theme for your editor and click “Next”

13. Click “Finish”

14. Once the HAXM installer finishes you should get a dialog box from Android Studio.

15. Click “AVD Manager”

16. Click “Create New Virtual Device”

17. Select a Device from the list and click “Next”

18. If the System Image for the most recent SDK is not installed, you will need to click the "Download" link next to the SDK version.

19. After the image has been downloaded, click “Next”

20. Preview the settings and then click “Finish”

21. You should see the newly created device in the “Your Virtual Device” screen.

22. Click the Play ▶button all the way to the right in the "Actions" column.

23. This will open an Android simulator on your machine. If everything is working well, it should load a generic screen with a few apps.

24. Go to your terminal and in your React Native project directory, run `npm start`

25. When the Metro Bundler opens up in your browser, click on the option to "Run on Android simulator". If it does not open in your browser, you may need to go to `localhost:19002` in your browser.

26. If you navigate back to your Android simulator you might see a "Permit Drawing Over Other Apps" message. Click the OK button and it should take you to the system settings screen to enable this. You will likely have to back to the Metro Bundler and select "Run on Android simulator" again.

27. This will launch the Expo application in the simulated device and run your bundled code. Eventually, it should load the "Hi There!" screen.