

Troubleshooting

These are some common issues you may run into while setting up React Native. If you encounter something that is not listed here, try [searching for the issue in GitHub](#).

Port already in use

The [Metro bundler](#) runs on port 8081. If another process is already using that port, you can either terminate that process, or change the port that the bundler uses.

Terminating a process on port 8081

Run the following command to find the id for the process that is listening on port 8081:

```
sudo lsof -i :8081
```

Then run the following to terminate the process:

```
kill -9 <PID>
```

On Windows you can find the process using port 8081 using [Resource Monitor](#) and stop it using Task Manager.

Using a port other than 8081

You can configure the bundler to use a port other than 8081 by using the `port` parameter, from the root of your project run:

npm Yarn

```
npm start -- --port=8088
```

You will also need to update your applications to load the JavaScript bundle from the new port. If running on device from Xcode, you can do this by updating occurrences of 8081 to your chosen port in the `ios/___App_Name___.xcodproj/project.pbxproj` file.

NPM locking error

If you encounter an error such as `npm WARN locking Error: EACCES` while using the React Native CLI, try running the following:

```
sudo chown -R $USER ~/.npm
sudo chown -R $USER /usr/local/lib/node_modules
```

Missing libraries for React

If you added React Native manually to your project, make sure you have included all the relevant dependencies that you are using, like `RCTText.xcodproj`, `RCTImage.xcodproj`. Next, the binaries built by these dependencies have to be linked to your app binary. Use the `Linked Frameworks and Binaries` section in the Xcode project settings. More detailed steps are here: [Linking Libraries](#).

If you are using CocoaPods, verify that you have added React along with the subspecs to the `Podfile`. For example, if you were using the `<Text />`, `<Image />` and `fetch()` APIs, you would need to add these in your `Podfile`:

```
pod 'React', :path => '../node_modules/react-native', :subspecs => [
  'RCTText',
  'RCTImage',
  'RCTNetwork',
  'RCTWebSocket',
]
```

Next, make sure you have run `pod install` and that a `Pods/` directory has been created in your project with React installed. CocoaPods will instruct you to use the generated `.xcworkspace` file henceforth to be able to use these installed dependencies.

React Native does not compile when being used as a CocoaPod

There is a CocoaPods plugin called [cocoapods-fix-react-native](#) which handles any potential post-fixing of the source code due to differences when using a dependency manager.

Argument list too long: recursive header expansion failed

In the project's build settings, `User Search Header Paths` and `Header Search Paths` are two configs that specify where Xcode should look for `#import` header files specified in the code. For Pods, CocoaPods uses a default array of specific folders to look in. Verify that this particular config is not overwritten, and that none of the folders configured are too large. If one of the folders is a large folder, Xcode will attempt to recursively search the entire directory and throw above error at some point.

To revert the `User Search Header Paths` and `Header Search Paths` build settings to their defaults set by CocoaPods - select the entry in the Build Settings panel, and hit delete. It will remove the custom override and return to the CocoaPod defaults.

No transports available

React Native implements a polyfill for WebSockets. These [polyfills](#) are initialized as part of the `react-native` module that you include in your application through `import React from 'react'`. If you load another module that requires WebSockets, such as [Firebase](#), be sure to load/require it after `react-native`:

```
import React from 'react';
import Firebase from 'firebase';
```

Shell Command Unresponsive Exception

If you encounter a `ShellCommandUnresponsiveException` exception such as:

```
Execution failed for task ':app:installDebug'.
com.android.builder.testing.api.DeviceException:
com.android.ddmlib.ShellCommandUnresponsiveException
```

Try downgrading your Gradle version to 1.2.3 in `android/build.gradle`.

react-native init hangs

If you run into issues where running `npx react-native init` hangs in your system, try running it again in verbose mode and referring to [#2797](#) for common causes:

```
npx react-native init --verbose
```

When you're debugging a process or need to know a little more about the error being thrown, you may want to use the verbose option to output more logs and information to nail down your issue.

Run the following command in your project's root directory.

npm Yarn

```
npm run android -- --verbose
```

Unable to start react-native package manager (on Linux)

Case 1: Error "code":"ENOSPC","errno":"ENOSPC"

Issue caused by the number of directories inotify (used by watchman on Linux) can monitor. To solve it, run this command in your terminal window


```
echo fs.inotify.max_user_watches=582222 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p
```

Error: spawnSync ./gradlew EACCES

If you run into issue where executing `npm run android` or `yarn android` on macOS throws the above error, try to run `sudo chmod +x android/gradlew` command to make `gradlew`

files into executable.

Is this page useful?  

 Edit this page

*Last updated on **Jun 21, 2023***