Source Maps

Source maps allow you to map a transformed file back to the original source file. The main purpose of source maps is to aid debugging and help with investigating issues from release builds.

Without the source maps, when running into an error in the release build you will see a stacktrace like:

```
TypeError: Cannot read property 'data' of undefined
  at anonymous(app://index.android.bundle:1:4277021)
  at call(native)
  at p(app://index.android.bundle:1:227785)
```

With source maps generated, a stacktrace will include path, file name, and line number of the original source file:

```
TypeError: Cannot read property 'data' of undefined
  at anonymous(src/modules/notifications/Permission.js:15:requestNotificationPermission)
  at call(native)
  at p(node_modules/regenerator-runtime/runtime.js:64:Generator)
```

This allows you to triage release issues using a decipherable stacktrace.

Follow the instructions below to get started with source maps.

Enable source maps on Android

Hermes



Source maps are built in Release mode by default, unless hermesFlagsRelease is set. In that case source maps will have to be enabled.

To do so, ensure the following is set in your app's android/app/build.gradle file.

```
project.ext.react = [
    enableHermes: true,
    hermesFlagsRelease: ["-0", "-output-source-map"], // plus whichever flag was required
to set this away from default
]
```

If done correctly you should see the output location of the source map during Metro build output.

```
Writing bundle output to:,
android/app/build/generated/assets/react/release/index.android.bundle
Writing sourcemap output to:,
android/app/build/intermediates/sourcemaps/react/release/index.android.bundle.packager.map
```

Development builds do not produce a bundle and thus already have symbols, but if the development build is bundled you may use hermesFlagsDebug like above to enable source maps.

Enable source maps on iOS

Source maps are disabled by default. To enable them one has to define a SOURCEMAP_FILE environment variable.

In order to do so, within Xcode head to the Build Phase - "Bundle React Native code and images".

At the top of the file near the other exports, add an entry for SOURCEMAP_FILE to the preferred location and name. Like below:

```
export SOURCEMAP_FILE="$(pwd)/../main.jsbundle.map";
export NODE_BINARY=node
../node_modules/react-native/scripts/react-native-xcode.sh
```

If done correctly you should see the output location of the source map during Metro build output.

```
Writing bundle output to:,
Build/Intermediates.noindex/ArchiveIntermediates/application/BuildProductsPath/Release-
iphoneos/main.jsbundle
Writing sourcemap output to:,
Build/Intermediates.noindex/ArchiveIntermediates/application/BuildProductsPath/Release-
iphoneos/main.jsbundle.map
```

Manual Symbolication

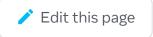


You may read about manual symbolication of a stack trace on the symbolication page.

Is this page useful?







Last updated on Jun 21, 2023