Debugging on Android and iOS

Estimated duration: 10 minutes

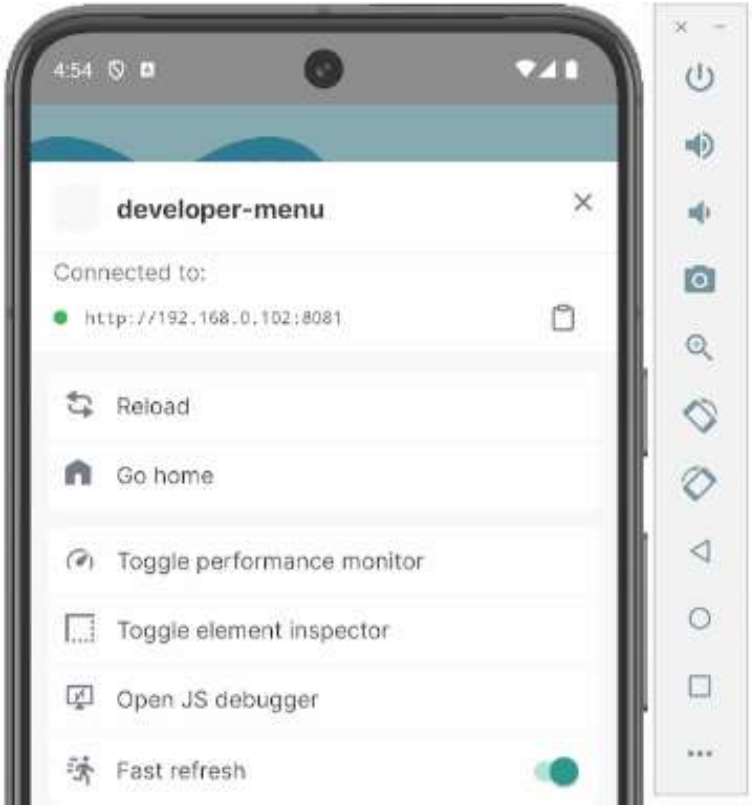
Debugging is crucial to the mobile development process, ensuring that applications run smoothly on different platforms. This reading focuses on the key debugging tools available for Android and iOS when developing React Native applications. These tools, such as the Element Inspector, Performance Monitor, and Chrome DevTools, help developers inspect UI elements, monitor app performance, and debug JavaScript code efficiently. By understanding how to use these tools, you can streamline your development and troubleshooting efforts.

Element Inspector: One click away

The Element Inspector is a powerful tool that functions similarly to the Chrome Inspector, allowing you to inspect UI components directly on your device or emulator. To open the Developer Menu and access the Element Inspector, follow these steps:

- **Physical device:** Shake the device.
- **iOS simulator:** Press `Cmd-Ctrl-Z` on macOS.
- **Android emulator:** Press `Cmd-M on macOS` or `Ctrl-M` on Windows.

The following screen will appear on the phone.



Once the Developer Menu appears on the screen, you can click on any element to check its style properties.

Developer menu options

The Developer menu provides several useful options:

- **Copy link:** Copies the dev server address in dev client or exp:// link in Expo of your app.
- **Reload:** Reloads the app. Usually not necessary since Fast Refresh is enabled by default.
- **Go home:** Returns to the home screen of the dev client's or Expo Go app.
- **Toggle performance monitor:** Enables or disables the performance information about your app.
- **Toggle element inspector:** Enables or disables the element inspector overlay.
- **Open JS debugger:** Opens Chrome DevTools for apps using Hermes, giving access to Console, Source, and Network tabs.

Element inspector in Expo

Performance Monitor: Heap, CPU, and Frame Rate

The Performance Monitor is a helpful tool for tracking the real-time performance of your app. To access it, open the Developer Menu and toggle the Performance Monitor option. Once enabled, a small table appears on the screen, displaying the following indicators:

- **RAM Usage:** Shows the amount of memory your app is consuming.
- **JavaScript Heap:** Helps identify memory leaks in your app.

- **Number of Views:** Displays two numbers— the top shows the total number of views on the screen, and the bottom shows the number of views in the component.
- **Frames Per Second (FPS):** Displays the FPS for both the UI thread used for native rendering and the JS thread where most of the app logic, such as API calls and touch events, is handled.

Performance monitor in Expo

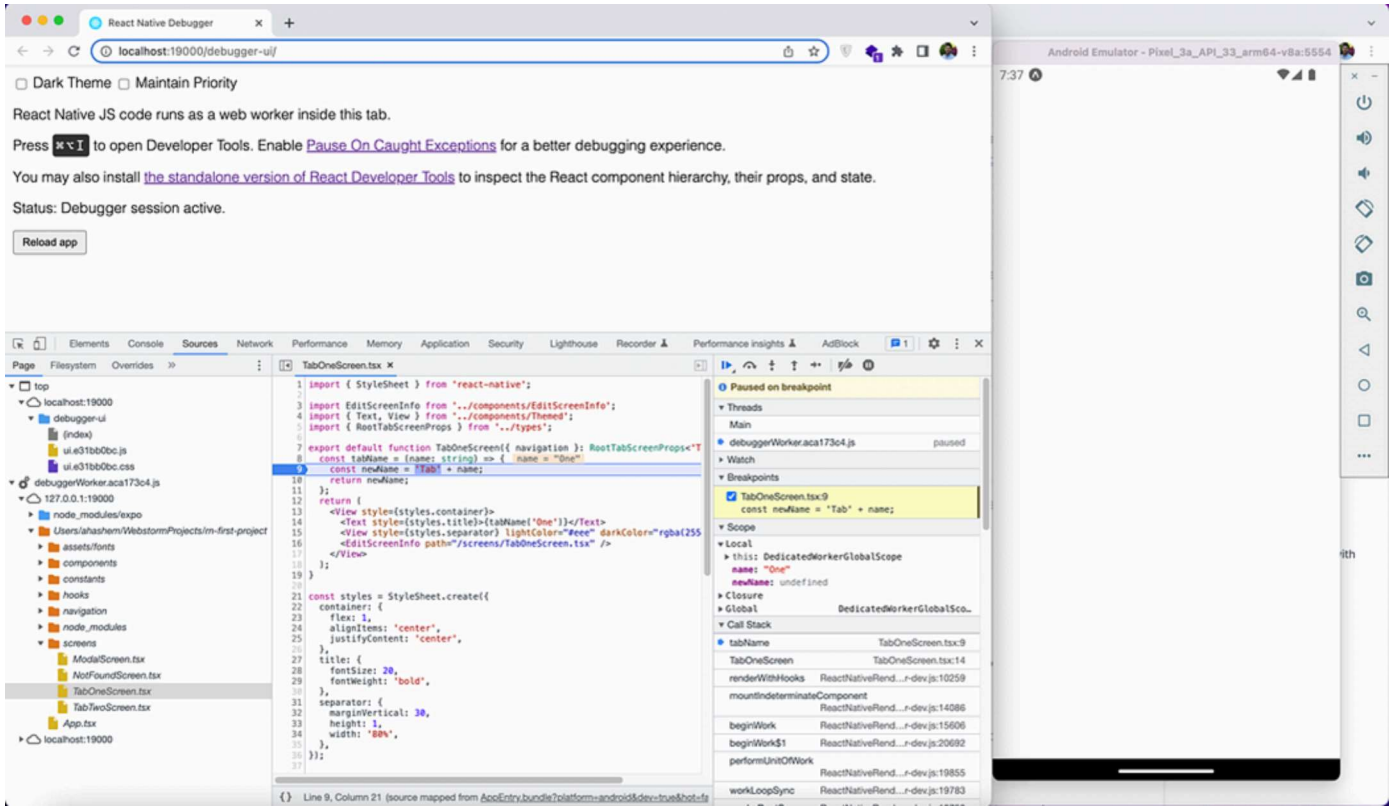
Remote Debugging with Chrome Dev Tools: No Setup Needed

Remote debugging allows you to use Chrome DevTools to inspect and debug JavaScript code running in your app. After running your project with `expo start`, you can launch the emulator by typing `i` for iOS or `a` for Android. If you haven't set up your local environment yet, you can check out this article for getting everything ready.

Once the app is running, follow these steps to enable remote debugging:

- **Physical device:** Shake the device.
- **iOS simulator:** Press `Ctrl-Cmd-Z`.
- **Android emulator:** Press `Cmd-M` on macOS or `Ctrl-M` on Windows.

Next, select **Debug Remote JS**. This will open the React Native Debugger interface at http://localhost:19000/debugger-ui/. You can then open Chrome DevTools and navigate to the **Sources tab**. In the navigator, you'll find **debuggerWorker/127.0.0.1:1999/path-to-your-project**.



Here, you can navigate through your project files and start debugging. For example, you can debug your `tabName` function by placing a breakpoint there.

Summary

Using these tools enhances the debugging process, ensuring better app performance and user experience.