



React Native CLI Vs EXPO

Filter by Topic



React Native CLI Vs EXPO

React Native CLI

1. Overview:

- **React Native CLI** is the original method to create React Native apps. It provides a more flexible and customizable environment.

2. Features:

- **Native Code Access:** Direct access to native code allows for greater flexibility and customization.
- **Custom Native Modules:** You can add custom native modules or third-party libraries that require native code.
- **Advanced Configuration:** More control over the configuration of Android and iOS builds.
- **Complex Integrations:** Suitable for projects requiring complex integrations with native SDKs or extensive native code.

3. Setup and Configuration:

- **Installation:** Requires setting up a development environment with Android Studio (for Android) and Xcode (for iOS, macOS only).
- **Customization:** High level of customization for native code and build configurations.

4. Pros:

- **Full Control:** Provides full control over the app's native code and configuration.

- **Flexibility:** Allows for custom native code and third-party libraries.
- **Performance:** May offer better performance for apps with complex native interactions.

5. Cons:

- **Complexity:** More complex setup and configuration compared to Expo.
- **Maintenance:** Requires managing native code and dependencies manually.

Expo CLI

1. Overview:

- **Expo CLI** is a framework and platform for universal React applications. It simplifies the development process with a managed workflow.

2. Features:

- **Managed Workflow:** Provides a managed environment with pre-configured settings and libraries.
- **Easy Setup:** Simplifies setup with no need for native code configuration.
- **Expo Go App:** Allows you to run and test your app on physical devices without installing a development environment.
- **Over-the-Air (OTA) Updates:** Supports pushing updates directly to users without needing to go through app store reviews.

3. Setup and Configuration:

- **Installation:** Easier setup with the Expo CLI and does not require Android Studio or Xcode (for basic development).
- **Managed Build:** Limited control over native code; focuses on a set of pre-configured libraries and components.

4. Pros:

- **Simplicity:** Easier to get started with a pre-configured environment.
- **Rapid Development:** Faster development with less configuration and setup.
- **Cross-Platform:** Provides tools for cross-platform development and testing.

5. Cons:

- **Limited Customization:** Less flexibility in adding custom native modules or libraries.
- **Performance:** May have performance limitations for apps requiring extensive native code or custom native modules.
- **Expo SDK Limitations:** Depends on the Expo SDK; some native libraries might not be supported.

When to Use Which

- **Use React Native CLI if:**
 - You need full control over native code and configuration.
 - Your app requires custom native modules or third-party libraries with native dependencies.
 - You are working on a complex project with specific native requirements.
- **Use Expo CLI if:**
 - You want a simpler, more streamlined development process.
 - You are building a prototype or a project that fits within the capabilities of the Expo SDK.
 - You want to avoid dealing with native code and complex configurations.
 -

Both tools have their strengths and are suitable for different types of projects. Choosing the right tool depends on your specific needs and the level of customization required.

