React Native Bare Workflow Project Structure

Full native code access. Run native modules and control iOS/Android projects directly.

#react-native #bare #mobile #ios #android #native

PNG PDF Copy Prompt

Project Directory

```
MyApp/
    index.js  App registry en…
    App.tsx  Root component
    package.json
    tsconfig.json
    babel.config.js
    metro.config.js  Metro bundler c…
    .gitignore
    Gemfile  Ruby deps for i…
  › android/  Android native …
      app/  Main Android app
      build.gradle
      settings.gradle
      gradle.properties
  › ios/  iOS native proj…
      MyApp/  Xcode project f…
      MyApp.xcworkspace
      Podfile  CocoaPods depen…
      Podfile.lock
  › src/  JavaScript/Type…
    › screens/  Screen componen…
        HomeScreen.tsx
        DetailsScreen.tsx
    › components/  Reusable compon…
        Button.tsx
        Card.tsx
    › navigation/  React Navigatio…
        AppNavigator.tsx  Root navigator
        types.ts  Navigation types
    › hooks/
        useAuth.ts
    › utils/
        storage.ts
  › __tests__/  Jest tests
      App.test.tsx
```

Why This Structure?

Bare workflow gives you full access to native iOS and Android projects. You can add any native module, customize build settings, and integrate native SDKs. Essential when Expo's managed workflow can't meet your needs.

Key Directories

`android/` - Full Android Studio project—modify Gradle, add native modules

`ios/` - Xcode project with Podfile for native dependencies

`src/` - All JavaScript/TypeScript code lives here

`src/navigation/` - React Navigation configuration and types

`src/screens/` - Screen components mapped to routes

Getting Started

1. `npx react-native@latest init MyApp`
2. `cd MyApp`
3. `npx pod-install` # Install iOS CocoaPods
4. `npx react-native run-ios` # or run-android

When To Use This

- Need native modules not supported by Expo
- Integrating native SDKs (payments, maps, analytics)
- Require custom native code (Swift/Kotlin)
- Full control over build configuration
- Enterprise apps with specific native requirements

When To Upgrade

- App architecture becomes complex (consider feature-based)
- Multiple teams working on different features
- State management needs structure
- Need better code organization and scalability

Trade-offs

`Setup complexity` - Must configure Xcode, Android Studio, CocoaPods

`Build times` - Native builds are slower than Expo Go preview

`Maintenance` - Must handle native upgrades and breaking changes