## **DEVELOPMENT WORKFLOW**

### **5.1 Running the App**

#### **Start Metro Bundler**

Metro is React Native’s JavaScript bundler — it watches for changes and reloads the app.

yarn start

# or

npx react-native start

#### **Run on Android**

npx react-native run-android

* Requires a running Android emulator (emulator -avd <name>)  
  or a USB-connected device with USB Debugging enabled.

#### **Run on iOS (macOS only)**

npx react-native run-ios

* Uses the default iPhone simulator unless you specify:

npx react-native run-ios --simulator="iPhone 15"

### **5.2 Hot Reload & Fast Refresh**

React Native supports Fast Refresh — instant UI updates when saving code.

* Enable it: Shake device → Developer Menu → “Enable Fast Refresh”
* Hot Reload vs. Fast Refresh:
  + Hot Reload → tries to patch code without losing state (deprecated in favor of Fast Refresh)
  + Fast Refresh → more reliable, can keep component state for most changes

### **5.3 Debugging JavaScript**

React Native has built-in debugging options:

* Flipper (recommended) → Desktop app for debugging RN apps
  + Install: [https://fbflipper.com](https://fbflipper.com/)
  + Includes logs, network inspector, React DevTools
* Console Logs → Use console.log, console.warn, console.error  
  (but clean them before release)

### **5.4 Debugging Native Build Errors**

* Android:
  + Run Gradle build manually:

cd android && ./gradlew assembleDebug

* + Clear Gradle cache if stuck:

cd android && ./gradlew clean

* iOS:
  + Open ios/MyApp.xcworkspace in Xcode
  + Clean build folder (Shift + Command + K)
  + Rebuild (Command + B)

### **5.5 Using React Native CLI Tools**

* Link assets:

npx react-native-asset

* Check environment:

npx react-native doctor

* Clear Metro cache (fixes weird reload issues):

npx react-native start --reset-cache

### **5.6 Device Testing**

* Android:
  + adb devices → lists connected devices
  + Logcat debugging:

adb logcat \*:S ReactNative:V ReactNativeJS:V

* iOS:
  + Test on real device → Requires Apple developer account & provisioning profile

### **5.7 Handling Common Dev-Time Errors**

* Red screen error → JavaScript runtime error, fix in code
* Build failed → Usually missing native dependency or wrong version
* Metro stuck → Restart with Ctrl+C and npx react-native start --reset-cache

✅ At the end, you can:

* Run & reload your app quickly
* Debug JS & native errors
* Test on emulator, simulator, or real device
* Use CLI tools to fix common problems