ANDROID | DEVELOPER GUIDE

QRS App — Build & Install Guide

Everything needed to connect a phone over Wi-Fi and install the app: **debug** builds for development with live reload, and **release** builds that run on their own.

Debug or release — which one?

	Debug APK	Release APK
Purpose	Day-to-day development	Testing, demos, sharing with others
Needs Metro running?	**Yes** — the code is downloaded from the PC	**No** — the code is inside the APK
Live reload (press r)	Yes	No
Developer menu	Yes (shake the phone)	No
Build time	~30-60 s	~2-5 min (longer after a clean)
Typical problem	White screen when Metro cannot be reached	None — but rebuild to see code changes
Build command	`gradlew assembleDebug`	`gradlew assembleRelease`

Rule of thumb: use **debug** while writing code, and **release** whenever the app must simply work — on a colleague's phone, in a demo, or when Wi-Fi debugging is unreliable.

Connecting the phone over Wi-Fi

Both the phone and the PC must be on the **same Wi-Fi network**. On the phone, enable **Settings** → **Developer options** → **Wireless debugging**.

	Pairing popup	Main wireless debugging screen
Where	"Pair device with pairing code" — opens a popup	The Wireless debugging screen itself
Shows	A 6-digit code + its own IP:PORT	"IP address & Port" — a DIFFERENT port
Used for	`adb pair` — once per PC	`adb connect` — every session
Expires	About 60 seconds — be quick	Changes whenever wireless debugging restarts

The most common mistake: using the **pairing** port for `adb connect`. The two ports are always different. Always read the port from the main screen immediately before connecting.

1. Connect the phone (do this first, every time)

1 Restart the adb server on the PC

```
adb kill-server
adb start-server
```

2 Pair the PC with the phone (first time only, or after "Forget")

On the phone tap **Pair device with pairing code** and keep the popup open. Use the IP:PORT shown **in the popup**:

```
adb pair 10.1.1.106:39073

# Enter pairing code: 369462
# Successfully paired to 10.1.1.106:39073
```

The port and the code are different every time. Read them from the popup as you type.

3 Connect, using the port from the MAIN screen

```
adb connect 10.1.1.106:41235
```

Pairing alone does not connect the device — this step is always required.

4 Confirm the phone is listed

```
adb devices

# List of devices attached
# 10.1.1.106:41235     device
```

"device" = ready. "offline" or "unauthorized" = turn wireless debugging off and on, then connect again with the new port.

If several devices are listed

An old or stale entry (for example `emulator-5554`) makes commands fail with *"more than one device"*. Remove it, or name the phone explicitly with `-s`:

```
adb disconnect 10.1.1.106:36827
adb -s 10.1.1.106:41235 install -r app-release.apk
```

2. Release build — the app runs on its own

Use this for demos, for other people's phones, and whenever Wi-Fi debugging is unreliable. The JavaScript is packed into the APK during the build.

1 Remove any leftover bundle (important)

```
del C:\QRSAPP\android\app\src\main\assets\index.android.bundle
```

If this file exists, Gradle ships it instead of building fresh code — the app then shows OLD screens.

2 Build

```
cd C:\QRSAPP\android
.\gradlew assembleRelease
```

Output: C:\QRSAPP\android\app\build\outputs\apk\release\app-release.apk

3 Check the APK is new

```
dir app\build\outputs\apk\release\app-release.apk
```

The timestamp must be from a few minutes ago.

4 Install

```
adb devices
adb uninstall com.qrsapp
adb install app\build\outputs\apk\release\app-release.apk
```

"uninstall" may report Failure [DELETE_FAILED_INTERNAL_ERROR] if the app is not installed — that is harmless.

> **Release APK not showing the latest changes?** In order: (1) delete `index.android.bundle` from `app\src\main\assets`, (2) check the source files really were replaced (look at their dates), (3) rebuild and confirm the APK timestamp.

Sharing the APK without adb

- Send **app-release.apk** by WhatsApp, email or Google Drive.
- On the phone, tap the file and allow **"Install unknown apps"** for the app used to open it.
- Uninstall the old QRS app first if Android refuses to replace it.

3. Debug build — development with live reload

The debug APK does **not** contain the app code. It downloads it from Metro on the PC every time it starts, which is what makes live reload possible.

1 Remove any packed bundle and stop old Metro processes

```
del C:\QRSAPP\android\app\src\main\assets\index.android.bundle
taskkill /F /IM node.exe
```

Several Metro processes fighting over port 8081 is a very common cause of white screens.

2 Start one Metro — terminal 1, leave it open

```
cd C:\QRSAPP
npx react-native start --reset-cache
```

It must say "Dev server ready" on port 8081 and must NOT offer port 8082.

3 Build and install — terminal 2

```
cd C:\QRSAPP\android
.\gradlew assembleDebug
adb install -r app\build\outputs\apk\debug\app-debug.apk
```

4 Point the app at Metro

```
adb reverse tcp:8081 tcp:8081
```

Must print 8081. Re-run it every time the phone reconnects or the app is reinstalled.

5 Open the app — Metro should print BUNDLE ./index.js

If the screen stays white

Set the bundle address on the phone. This survives reconnects, unlike `adb reverse`:

- Open the developer menu: shake the phone, or run `adb shell input keyevent 82`.
- Tap **Change Bundle Location** (or Settings → Debug server host & port).
- Type the PC's Wi-Fi IP and port, e.g. **10.1.1.114:8081** — no http://, no spaces.
- Go back, then **Reload**.

Find the PC's IP with `ipconfig` (IPv4 Address on the Wi-Fi adapter). Test from the phone's browser: `http://10.1.1.114:8081/status` should show *packager-status:running*.

4. Troubleshooting

Message / symptom	Cause	Fix
`failed to connect` / `actively refused`	Wrong port, or the PC is no longer paired	Read the port from the main screen again; pair again if "Paired devices" is empty
`device offline`	Wireless link dropped	Toggle wireless debugging off/on, then connect with the new port
`more than one device/emulator`	A stale emulator entry	`adb disconnect <id>`, or add `-s <phone>` to every command
`No online devices found`	Phone dropped during the build	Reconnect, then install the APK directly with `adb install`
`INSTALL_FAILED_UPDATE_INCOMPATIBLE`	Debug and release are signed differently	`adb uninstall com.qrsapp`, then install again
`Another process is running on port 8081`	Old Metro still running	`taskkill /F /IM node.exe`, then start one Metro
White screen, "No apps connected" in Metro	The app cannot reach Metro	Re-run `adb reverse`, or set Change Bundle Location on the phone
CMake / ninja error on build	Stale native cache after a clean	Delete `app\.cxx`, `app\build`, `build`, then build again (no `clean`)
Build sits at 95% for a long time	Native C++ compile after a clean	Normal — allow 10-25 minutes; do not interrupt it

Deleting caches safely (PowerShell)

```
cd C:\QRSAPP\android
Remove-Item -Recurse -Force app\.cxx, app\build, build -ErrorAction SilentlyContinue
.\gradlew assembleRelease
```

PowerShell does not accept the old "rmdir /s /q" syntax.

Quick reference

Task	Command
Pair (first time)	`adb pair 10.1.1.106:<pairing port>`
Connect	`adb connect 10.1.1.106:<main port>`
List devices	`adb devices`
Build debug	`cd C:\QRSAPP\android ; .\gradlew assembleDebug`
Build release	`cd C:\QRSAPP\android ; .\gradlew assembleRelease`
Install	`adb install -r <path to .apk>`
Uninstall	`adb uninstall com.qrsapp`

Task	Command
Link app to Metro	`adb reverse tcp:8081 tcp:8081`
Open developer menu	`adb shell input keyevent 82`
See app logs	`adb logcat -s ReactNativeJS:*`