KINTO QA

Mobile App Development

Converting Web App to Android APK & iOS IPA

Using Capacitor by Ionic Framework

Version 1.0 | November 2025

Table of Contents

1. Introduction to Capacitor

Capacitor is a cross-platform native runtime by Ionic that allows you to convert your existing React web application into native mobile apps for Android (APK) and iOS (IPA) without rewriting your code.

Why Capacitor?

- Wraps your existing React app - no rewrite needed
- Access to native device features (camera, GPS, filesystem)
- Builds APK and IPA from the same codebase
- Native performance with WebView technology
- Active community and regular updates

How It Works

Capacitor wraps your web application in a native WebView component and provides JavaScript APIs to access native device features. Your existing React code runs unchanged inside the native app.

2. Prerequisites

For Android Development:

- Java JDK 11 or higher
- Android Studio (latest version)
- Android SDK (API level 22+)
- Node.js 16+ and npm
- ANDROID_HOME environment variable set

For iOS Development:

- macOS (required for iOS builds)
- Xcode 14 or higher
- CocoaPods package manager
- Xcode Command Line Tools
- Apple Developer account ($99/year for distribution)

Check Your Environment:

```
node --version    # Should be 16+
npm --version
java -version     # Should be 11+
```

3. Installing Capacitor

Step 1: Install Core Packages

```
npm install @capacitor/core @capacitor/cli
```

Step 2: Initialize Capacitor

```
npx cap init
```

You will be prompted for:

- App name: "KINTO QA"
- App ID: "com.kinto.qa" (reverse domain notation)
- Web directory: "dist/public" (your build output)

Step 3: Install Platform Packages

```
npm install @capacitor/android @capacitor/ios
```

Step 4: Add Platforms

```
npx cap add android
npx cap add ios
```

This creates android/ and ios/ folders with native projects.

4. Configuring Your Project

capacitor.config.ts

After initialization, you will have a capacitor.config.ts file:

```
import type { CapacitorConfig } from '@capacitor/cli';
const config: CapacitorConfig = {
  appId: 'com.kinto.qa',
  appName: 'KINTO QA',
  webDir: 'dist/public',
  server: {
    androidScheme: 'https'
  }
};
export default config;
```

Update API Endpoints

Configure your app to connect to your backend server:

```
const API_URL = import.meta.env.VITE_API_URL ||
  'https://your-production-server.com';
```

Build Your Web App

Before adding to mobile platforms, build your React app:

```
npm run build
```

Sync Web Assets

Copy web assets to native projects:

```
npx cap sync
```

5. Building Android APK

Method 1: Using Android Studio (Recommended)

Step 1: Open Android Studio

```
npx cap open android
```

Step 2: Wait for Gradle sync to complete

Step 3: Build Debug APK

- Go to Build !’ Build Bundle(s) / APK(s) !’ Build APK(s)
- Wait for build to complete
- APK location: android/app/build/outputs/apk/debug/app-debug.apk

Step 4: Build Release APK (Production)

- Go to Build !’ Generate Signed Bundle / APK
- Select APK and click Next
- Create or select existing keystore
- Fill in keystore details and build

Method 2: Command Line

```
cd android
./gradlew assembleDebug      # Debug APK
./gradlew assembleRelease    # Release APK
```

Creating a Keystore

For production release, create a keystore:

```
keytool -genkey -v -keystore kinto-qa.keystore \
  -alias kinto-qa -keyalg RSA -keysize 2048 \
  -validity 10000
```

Store the keystore file and password securely!

6. Building iOS IPA

& þ Important: iOS builds require macOS with Xcode installed

Step 1: Install CocoaPods

```
sudo gem install cocoapods
pod repo update
```

Step 2: Open Xcode

```
npx cap open ios
```

Step 3: Configure Signing

In Xcode:

- Select your project in the left sidebar
- Go to Signing & Capabilities tab
- Select your Team (requires Apple Developer account)
- Choose Automatically manage signing

Step 4: Build for Testing (Simulator)

- Select a simulator device
- Click Product !' Run (or press Cmd+R)

Step 5: Build IPA for Distribution

For App Store or Ad Hoc distribution:

- Connect a real device or select Generic iOS Device
- Click Product !' Archive
- Wait for archive to complete
- In Organizer, click Distribute App
- Choose distribution method:
    - App Store: For public distribution
    - Ad Hoc: For internal testing (max 100 devices)
    - Enterprise: For in-house distribution (requires Enterprise account)

Step 6: Export IPA

- Follow the wizard to export IPA file
- IPA will be saved to your chosen location

7. Adding Native Features

Camera Access

For capturing photos in checklists:

```
npm install @capacitor/camera
```

Usage example:

```
import { Camera, CameraResultType } from '@capacitor/camera';

const takePicture = async () => {
  const image = await Camera.getPhoto({
    quality: 90,
    allowEditing: false,
    resultType: CameraResultType.Uri
  });
  return image.webPath;
};
```

Barcode Scanner

For scanning equipment barcodes:

```
npm install @capacitor-community/barcode-scanner
```

File System

For offline data storage:

```
npm install @capacitor/filesystem
```

Push Notifications

For maintenance alerts:

```
npm install @capacitor/push-notifications
```

Device Info

For detecting platform:

```
npm install @capacitor/device
```

```
import { Device } from '@capacitor/device';
const info = await Device.getInfo();
console.log('Platform:', info.platform);
```

8. Development Workflow

Live Reload During Development

Configure live reload in capacitor.config.ts:

```
const config: CapacitorConfig = {
  // ... other config
  server: {
    url: 'http://192.168.1.100:5000',  // Your local IP
    cleartext: true
  }
};
```

Testing on Emulators/Simulators

```
npx cap run android    # Run on Android emulator
npx cap run ios        # Run on iOS simulator
```

Regular Development Flow

1. Make changes to your React code
2. Build your web app: npm run build
3. Sync to native: npx cap sync
4. Test on device/emulator

Debugging

Android - Use Chrome DevTools:

- Open chrome://inspect in Chrome
- Select your device
- Click Inspect

iOS - Use Safari Web Inspector:

- Enable Web Inspector in iOS Settings !' Safari !' Advanced
- Open Safari !' Develop !' [Your Device]
- Select your app

9. Troubleshooting

Android Issues

Java Version Error:

```
java -version  # Check version (should be 11+)
export JAVA_HOME=/path/to/jdk-11
```

Gradle Build Failed:

```
cd android
./gradlew clean
./gradlew build --stacktrace
```

SDK Not Found:

```
export ANDROID_HOME=/path/to/Android/sdk
export PATH=$PATH:$ANDROID_HOME/tools
```

iOS Issues

CocoaPods Error:

```
cd ios/App
pod deintegrate
pod install
```

Code Signing Error:

- Verify Apple Developer account is active
- Check Team selection in Xcode
- Try Automatically manage signing

Build Failed:

```
xcodebuild clean
rm -rf ~/Library/Developer/Xcode/DerivedData/*
```

General Issues

White Screen on Launch:

- Check webDir path in capacitor.config.ts
- Verify build output exists in dist/public
- Run npx cap sync again

API Not Connecting:

- Update API_URL to production server
- Check CORS configuration on server
- Verify network permissions in AndroidManifest.xml

Appendix A: Quick Command Reference

Installation:

```
npm install @capacitor/core @capacitor/cli
npx cap init
npm install @capacitor/android @capacitor/ios
npx cap add android
npx cap add ios
```

Build & Sync:

```
npm run build              # Build web app
npx cap sync               # Sync to native projects
npx cap copy               # Copy web assets only
```

Open IDE:

```
npx cap open android       # Open Android Studio
npx cap open ios           # Open Xcode
```

Run on Device:

```
npx cap run android        # Run on Android
npx cap run ios            # Run on iOS
```

Build APK (Command Line):

```
cd android
./gradlew assembleDebug
./gradlew assembleRelease
```

Appendix B: Recommended Plugins for KINTO QA

Essential Plugins:

1. Camera
For capturing checklist photos
```
npm install @capacitor/camera
```

2. Barcode Scanner
For scanning equipment and parts
```
npm install @capacitor-community/barcode-scanner
```

3. Filesystem
For offline data storage
```
npm install @capacitor/filesystem
```

4. Network
For detecting offline/online status
```
npm install @capacitor/network
```

5. Device
For platform detection
```
npm install @capacitor/device
```

Optional Plugins:

- **Push Notifications**
  For maintenance alerts and reminders
- **Geolocation**
  For tracking inspection locations
- **Local Notifications**
  For offline PM reminders

End of Document

For more information, visit: https://capacitorjs.com/docs
Capacitor is developed by the Ionic team