

Module 11: App Deployment and Publishing

1. App release process for iOS and Android

iOS (App Store)

1. Set app version + build number in pubspec.yaml and Xcode.
2. Switch Flutter build mode to **release**.
3. Build an iOS archive using Xcode.
4. Upload the archive to **App Store Connect**.
5. Fill app details, screenshots, and app info.
6. Submit the app for **App Review**.
7. After approval, release it to users.

Android (Google Play)

1. Set app version + build number in pubspec.yaml.
2. Create a **release keystore** and sign the app.
3. Build the **AAB** (required by Google Play).
4. Upload AAB to **Google Play Console**.
5. Fill store listing, screenshots, and content rating.
6. Send app for review.
7. Publish after approval.

2. Steps to generate app bundles and APKs in Flutter

Android

- **AAB (recommended for Play Store):**
- flutter build appbundle --release

Output: build/app/outputs/bundle/release/app-release.aab

- **APK (for manual install/testing):**
- flutter build apk --release

Output: build/app/outputs/flutter-apk/app-release.apk

iOS

- Run:
- flutter build ios --release
- Then open Xcode → **Product > Archive** to generate the final release build for upload.

3. Best practices for submitting apps

App Store (iOS)

- Use real device testing before submission.
- Provide accurate screenshots and descriptions.
- Make sure app follows **Apple guidelines**.
- Use strong app signing & correct bundle IDs.
- Keep version numbers consistent.

Google Play (Android)

- Upload **AAB**, not APK, for store release.
- Add correct app permissions + privacy policy.
- Pass Google Play's **content rating**.
- Provide high-quality screenshots + icon.
- Test on different Android devices.