Video Converter Android - Build Configuration

Release Keystore Setup

This document describes how to set up the release keystore for the Video Converter Android app.

1. Generate Release Keystore

For production builds, you need to generate a release keystore:

```
# Navigate to android/app directory
cd android/app

# Generate keystore (replace with your details)
keytool -genkey -v -keystore my-upload-key.keystore -alias my-key-alias -keyalg
RSA -keysize 2048 -validity 10000

# You'll be prompted to enter:
# - Keystore password (remember this!)
# - Key password (remember this!)
# - Your name and organization details
```

2. Configure Gradle Properties

Create or edit `android/gradle.properties` and add:

```
# Release keystore configuration
MYAPP_UPLOAD_STORE_FILE=my-upload-key.keystore
MYAPP_UPLOAD_KEY_ALIAS=my-key-alias
MYAPP_UPLOAD_STORE_PASSWORD=your_keystore_password
MYAPP_UPLOAD_KEY_PASSWORD=your_key_password
```

Important: Never commit gradle.properties to version control! Add it to .gitignore.

3. Build Commands

Debug Build

```
# Build debug APK
npm run build:android:debug
# Or directly with gradle
cd android && ./gradlew assembleDebug
```

Release Build

```
# Build release APK
npm run build:android:release
# Or directly with gradle
cd android && ./gradlew assembleRelease
```

Bundle for Play Store

```
# Build Android App Bundle (AAB) for Play Store
cd android && ./gradlew bundleRelease
```

4. APK Output Locations

After building, APKs will be located at:

Debug:

- `android/app/build/outputs/apk/debug/app-debug.apk`

Release (Split APKs):

- `android/app/build/outputs/apk/release/app-armeabi-v7a-release.apk`
- `android/app/build/outputs/apk/release/app-arm64-v8a-release.apk`
- `android/app/build/outputs/apk/release/app-x86-release.apk`
- `android/app/build/outputs/apk/release/app-x86_64-release.apk`

Bundle:

- `android/app/build/outputs/bundle/release/app-release.aab`

5. Build Variants Explained

Split APKs by Architecture

- **armeabi-v7a**: 32-bit ARM (older devices)
- **arm64-v8a**: 64-bit ARM (most modern devices)
- **x86**: 32-bit Intel (emulators, some tablets)
- **x86_64**: 64-bit Intel (newer emulators, Intel devices)

Version Codes

Each architecture gets a unique version code:

- Base version: 1
- armeabi-v7a: 11

- arm64-v8a: 12
- x86: 13
- x86_64: 14

6. Performance Optimizations

The build configuration includes:

- **ProGuard/R8**: Code minification and obfuscation
- **Resource shrinking**: Removes unused resources
- **Split APKs**: Smaller downloads per architecture
- **FFmpeg optimization**: Proper native library packaging
- **React Native optimizations**: Development code removal

7. Testing Release Builds

Before publishing:

1. Test on physical devices (different architectures)
2. Test video conversion functionality
3. Verify app size is reasonable
4. Check crash reporting works
5. Test on Android 7.0+ (API 24+)

8. Troubleshooting

Build Fails

- Check keystore configuration in gradle.properties
- Ensure all dependencies are installed
- Clean build: `cd android && ./gradlew clean`

APK Too Large

- Verify split APKs are working
- Check if unused dependencies can be removed
- Consider using AAB format for Play Store

Video Processing Issues

- Ensure FFmpeg libraries are included
- Check ProGuard rules for video processing classes
- Test on target architecture devices

9. Security Notes

- Keep keystore file secure and backed up
- Never share keystore passwords
- Use different keystores for debug/release

- Consider using Play App Signing for additional security

10. CI/CD Integration

For automated builds, store keystore and passwords securely:

- Use encrypted environment variables
- Store keystore as base64 in CI secrets
- Automate signing process in build pipeline

Version Information

- **App Version**: 1.0.0
- **Version Code**: 1 (base)
- **Target SDK**: 34 (Android 14)
- **Min SDK**: 26 (Android 8.0)
- **Build Tools**: Latest stable