DigitalValut Chat - Deployment Guide

This guide provides step-by-step instructions for deploying DigitalValut Chat on Android and iOS devices.

Table of Contents

Prerequisites

Required Software

1. **Flutter SDK** (3.0+)
   - Download: https://docs.flutter.dev/get-started/install
   - Verify: `flutter --version`

2. **Git**
   - Download: https://git-scm.com/downloads
   - Verify: `git --version`

3. **Android Studio** (for Android)
   - Download: https://developer.android.com/studio
   - Install Android SDK and command-line tools

4. **Xcode** (for iOS, macOS only)
   - Download: https://developer.apple.com/xcode/
   - Install via Mac App Store

System Requirements

- **Windows**: Windows 10 or later (64-bit)
- **macOS**: macOS 10.15 (Catalina) or later
- **Linux**: Ubuntu 20.04 or later (64-bit)
- **RAM**: Minimum 8GB (16GB recommended)
- **Disk Space**: 10GB free space

Android Deployment

Step 1: Prepare Development Environment

```
# Install Flutter dependencies
flutter doctor

# Accept Android licenses
flutter doctor --android-licenses
```

Step 2: Connect Android Device

1. Enable Developer Options:
   - Settings → About Phone → Tap “Build Number” 7 times

2. Enable USB Debugging:
   - Settings → Developer Options → USB Debugging

3. Connect via USB cable

4. Verify connection:

```
flutter devices
```

Step 3: Build APK

Debug Build (for testing):

```
flutter build apk --debug
```

Release Build (for distribution):

```
flutter build apk --release
```

Split APK by ABI (smaller file size):

```
flutter build apk --release --split-per-abi
```

Output location: `build/app/outputs/flutter-apk/`

Step 4: Sign APK (for release)

1. Create keystore:

```
keytool -genkey -v -keystore ~/digitalvalut-release-key.jks -keyalg RSA -keysize 2048
-validity 10000 -alias digitalvalut
```

1. Create `android/key.properties`:

```
storePassword=YOUR_PASSWORD
keyPassword=YOUR_PASSWORD
keyAlias=digitalvalut
storeFile=/path/to/digitalvalut-release-key.jks
```

1. Update `android/app/build.gradle`:

```
def keystoreProperties = new Properties()
def keystorePropertiesFile = rootProject.file('key.properties')
if (keystorePropertiesFile.exists()) {
    keystoreProperties.load(new FileInputStream(keystorePropertiesFile))
}

android {
    ...
    signingConfigs {
        release {
            keyAlias keystoreProperties['keyAlias']
            keyPassword keystoreProperties['keyPassword']
            storeFile keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null
            storePassword keystoreProperties['storePassword']
        }
    }
    buildTypes {
        release {
            signingConfig signingConfigs.release
        }
    }
}
```

1. Build signed APK:

```
flutter build apk --release
```

Step 5: Install on Device

Via USB:

```
flutter install
```

Manual Installation:
1. Copy APK to device
2. Open file manager
3. Tap APK file
4. Allow “Install from Unknown Sources”
5. Tap “Install”

iOS Deployment

Step 1: Prepare Development Environment (macOS only)

```
# Install CocoaPods
sudo gem install cocoapods

# Verify Xcode installation
xcode-select --install

# Accept Xcode license
sudo xcodebuild -license accept
```

Step 2: Configure Signing

1. Open project in Xcode:

```
open ios/Runner.xcworkspace
```

1. Select Runner in project navigator
2. Go to “Signing & Capabilities”
3. Select your development team
4. Choose a unique Bundle Identifier

Step 3: Connect iOS Device

1. Connect iPhone/iPad via USB
2. Trust computer on device
3. Verify connection:

```
flutter devices
```

Step 4: Build for Device

Debug Build:

```
flutter build ios --debug
```

Release Build:

```
flutter build ios --release
```

Step 5: Deploy to Device

Option 1: Via Flutter

```
flutter run --release
```

Option 2: Via Xcode
1. Open `ios/Runner.xcworkspace`

2. Select your device
3. Click Run (▶️) button

Option 3: Create IPA for Distribution
1. Open Xcode
2. Product → Archive
3. Distribute App
4. Choose distribution method:
- Ad Hoc (for testing)
- App Store (for App Store)
- Development (for internal testing)
5. Export IPA

Step 6: Install IPA on Device

Using Xcode:
1. Window → Devices and Simulators
2. Select your device
3. Drag IPA to "Installed Apps"

Using TestFlight (recommended for beta):
1. Upload to App Store Connect
2. Invite testers
3. Testers install via TestFlight app

Troubleshooting

Common Issues

"Flutter command not found"

```
# Add to PATH (macOS/Linux)
export PATH="$PATH:/path/to/flutter/bin"

# Add to PATH (Windows)
# Add to System Environment Variables
```

"SDK location not found"

```
# Create android/local.properties
sdk.dir=/path/to/Android/sdk
flutter.sdk=/path/to/flutter
```

"CocoaPods not installed" (iOS)

```
sudo gem install cocoapods
pod setup
cd ios && pod install
```

"Build failed: Error signing"

- Verify Apple Developer account

- Check Bundle Identifier is unique
- Ensure provisioning profile is valid

“Device not found”

```
# Check connected devices
flutter devices

# Restart ADB (Android)
adb kill-server
adb start-server

# Restart device
```

Getting Help

- Documentation: https://docs.digitalvalut.chat
- Issues: https://github.com/yourusername/digitalvalut_chat/issues
- Community: https://discord.gg/digitalvalut
- Email: support@digitalvalut.chat

Best Practices

For Production Deployment

1. **Test thoroughly** on multiple devices
2. **Sign your builds** with secure keystores
3. **Keep keystores secure** - never commit to Git
4. **Version your releases** properly
5. **Create release notes** for users
6. **Test on different Android/iOS versions**

Security Considerations

1. **Use secure signing certificates**
2. **Enable ProGuard** for Android (code obfuscation)
3. **Test security features** (screenshot blocking, biometrics)
4. **Verify encryption** is working properly
5. **Review permissions** before release

Release Checklist

- [ ] All tests pass (`flutter test`)
- [ ] Code analyzed (`flutter analyze`)
- [ ] Tested on Android device
- [ ] Tested on iOS device (if applicable)
- [ ] Version updated in `pubspec.yaml`
- [ ] Release notes written

- [ ] APK/IPA signed properly
- [ ] Security features verified
- [ ] README updated
- [ ] CHANGELOG updated

Questions? Contact us at dev@digitalvalut.chat

Made with ❤️ by the DigitalValut Team