

Flutter App Release on Play Store

Step 1: Generate Key

Run the following command in your terminal (replace \$env:USERPROFILE with the actual path of your project's android/app directory):

bash

Copy code

```
keytool -genkey -v -keystore $env:USERPROFILE\upload-keystore.jks '
    -storetype JKS -keyalg RSA -keysize 2048 -validity 10000 '
    -alias upload
```

Set the key password and enter the project information as prompted. Make sure to note down the keystore password for future reference.

Step 2: Create key.properties

In the android folder of your project, create a new file named key.properties and add the following lines:

properties

Copy code

```
storePassword=<your-keystore-password>
keyPassword=<your-key-password>
keyAlias=upload
storeFile=<path-to-your-keystore-file>
```

Replace <your-keystore-password>, <your-key-password>, and <path-to-your-keystore-file> with the appropriate values.

Step 3: Configure build.gradle

In the android/app/build.gradle file, add the following lines at the top to load the keystore properties:

groovy

Copy code

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

Step 4: Configure Signing in build.gradle

In the same android/app/build.gradle file, modify the android block to include signing configurations for release builds:

groovy

Copy code

```
android {
    // Existing configuration...

    signingConfigs {
        release {
            keyAlias = keystoreProperties['keyAlias']
            keyPassword = keystoreProperties['keyPassword']
            storeFile = keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null
            storePassword = keystoreProperties['storePassword']
        }
    }

    buildTypes {
        release {
            signingConfig = signingConfigs.release
        }
    }
}
```

```
}  
}  
}
```

Step 5: Build APK and App Bundle

Update the app's package name (e.g., `com.example.project_name`) if needed.

To build an APK, run the following command:

bash

Copy code

```
flutter build apk
```

To build an App Bundle, run:

bash

Copy code

```
flutter build appbundle
```

Step 6: Upload to Google Play Console

Navigate to Google Play Console and follow the steps to set up your app:

Provide app information.

Set permissions.

Agree to terms and conditions.

Upload your APK or App Bundle and submit it for review.

Step 7: Post-Release Process

After approximately 7-8 days, your app should be published on the Play Store.

For future releases, remember to update the version number in `pubspec.yaml`. For example, if the current version is 0.0.4, update it to 0.0.4+1 or the next version before building and uploading the bundle again.