



# Android Studio

" How do we publish our app to the Play Store?  
What challenges await us? "

# INNEHÅLLSFÖRTECKNING

01

Översikt

03

Sign for  
Release

02

Google Play  
Recommendations

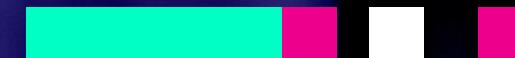
04

Uppgifter  
&  
Övningar

# 01

## ÖVERSIKT

# Produkt = REDO Lansering



# Lansering?

Varför en lektion på detta?

- Rekommendationer
  - Icon?
  - Packaging?
  - Project Signing?
  - Manifest.xml?
  - Cleaning?
  - Logging?



*Frågor?*

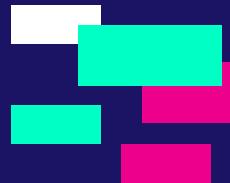
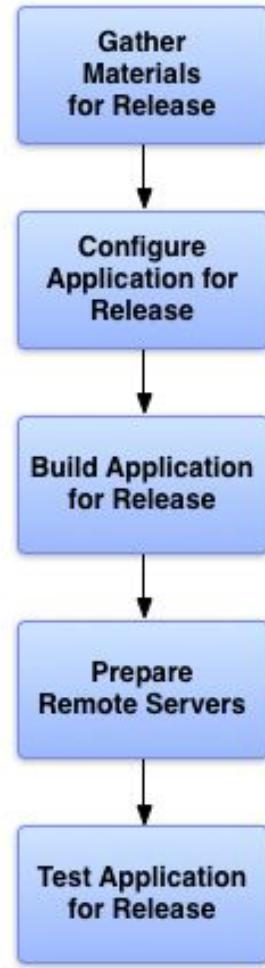


# 02

## Google Play Recommendations

# Main Tasks For Release





# ICON (Ikon) Rekommandationer

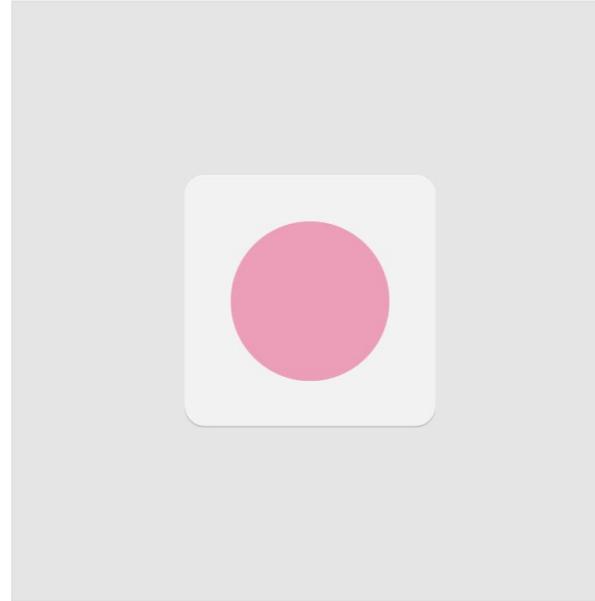


## Color

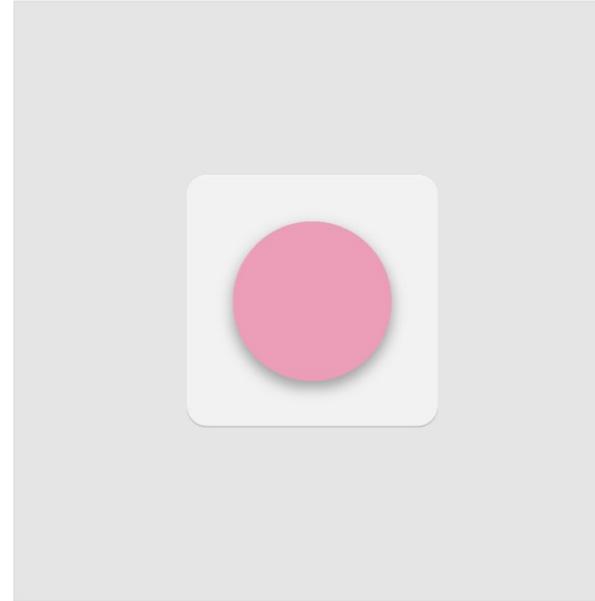


Material elements have tactile surfaces. When combined with color, these surfaces can produce numerous unique combinations.

Color is flush with an element's surface. Because color itself has no depth, it doesn't contain edges or shadows.



Do



Don't

## App icon

Your app's icon helps users identify your app on a device's Home screen and in the Launcher window. It also appears in Manage Applications, My Downloads, and elsewhere. In addition, publishing services such as Google Play display your icon to users. Be sure you have an app icon and that it meets the recommended [icon guidelines](#).



**Note:** If you are releasing your app on Google Play, you need to create a high-resolution version of your icon. See [Add preview assets to showcase your app](#) for more information.

## Rules...



When creating your artwork, ensure it conforms to the following:

- Final size: 512px x 512px
- Format: 32-bit PNG
- Color space: sRGB
- Max file size: 1024KB
- Shape: Full square – Google Play dynamically handles masking. Radius will be equivalent to 20% of icon size.
- Shadow: None – Google Play dynamically handles shadows. See 'Shadows' section below on including shadows within your artwork.

## Guidelines...



Wrong

Don't round the edges of your final asset



Don't

Don't use text or graphic elements to indicate ranking



Don't

Don't use text or graphic elements to indicate participation in a Play program



Right

Fill your entire asset with artwork when possible



Don't

Don't use text or graphic elements to promote deals or incentivize installs



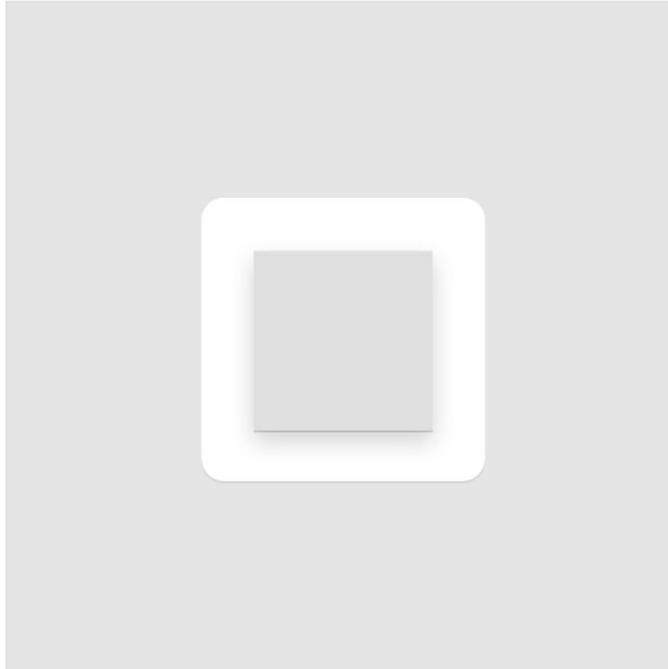
Don't

Don't use text or graphic elements that can mislead users

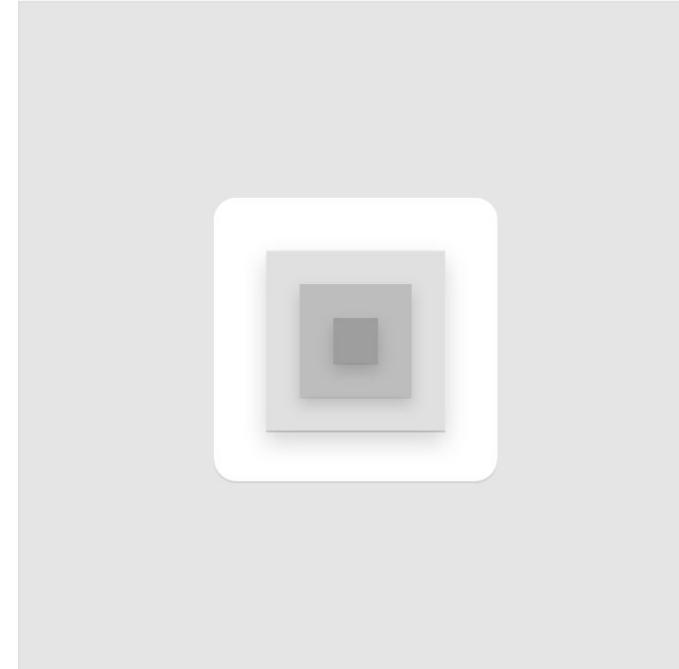
# Layer



When Material elements are layered, depth is produced by the shadows each layer casts. The number of surface layers that overlap should be limited, as too many layers can overcomplicate an icon.



Do



Don't



Don't add too many layers.

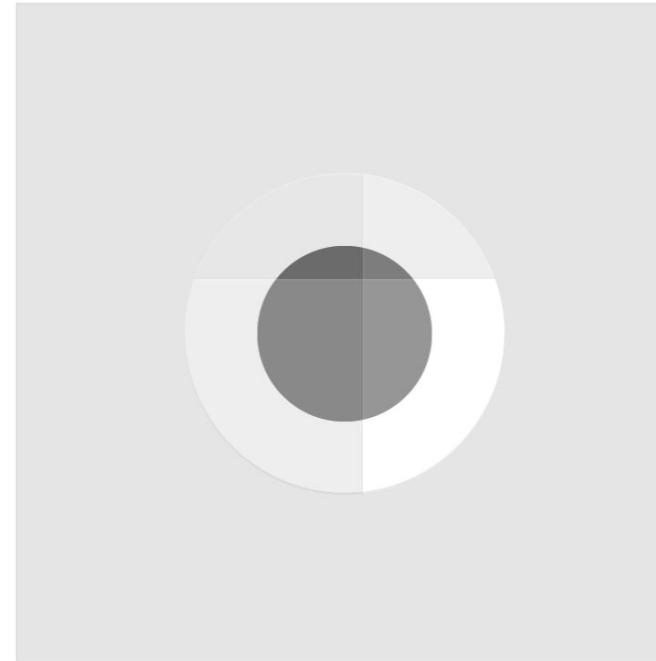


## Scoring

Scoring an icon creates the illusion of depth by dividing surfaces in half. Scores should be centered and placed on symmetrical shapes.



**Do**



**Don't**

Don't use multiple scores or position a score off-center.

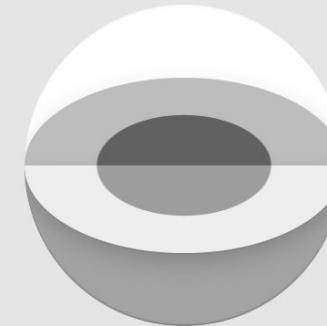
## Fold



When folded at multiple angles, Material elements have greater dimension.



Do



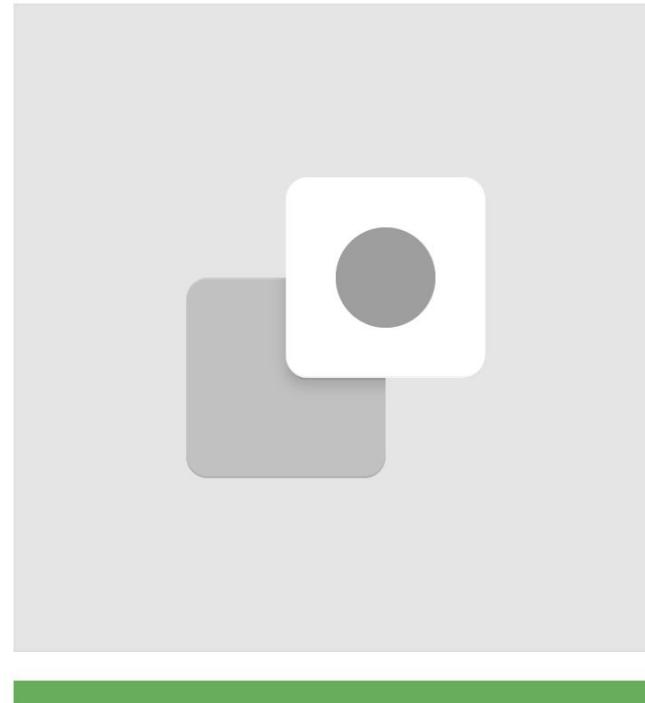
Don't

Don't use spot color on folded elements, to avoid altering or misrepresenting key elements.

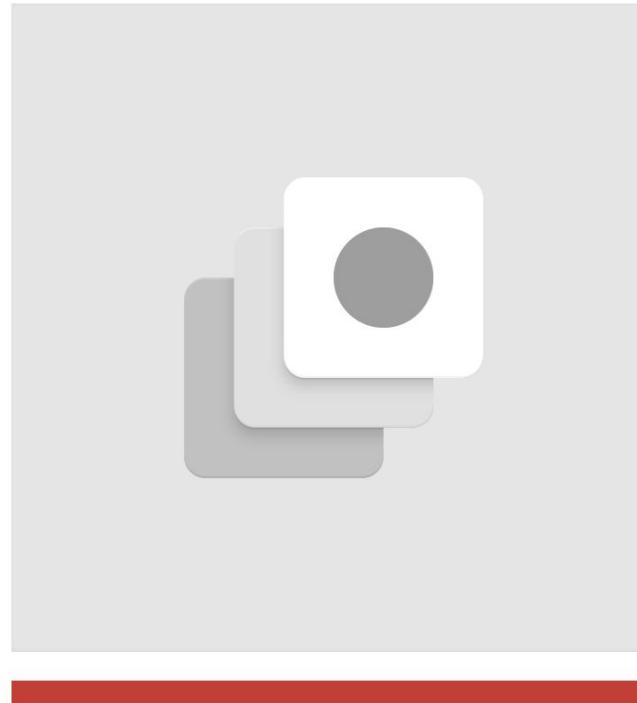
# Overlap



When Material elements overlap, it creates unique silhouettes. All elements, edges, and shadows are confined to the interior of the silhouette.



Do



Don't

Don't overlap more than two elements to avoid overcomplicating the icon.

# Accordion



An accordion fold involves adjoining two Material elements with a connecting fold, adding dimension to an element.



**Do**



**Don't**

Don't exceed more than two accordion folds, as having too many complicates the icon and doesn't provide a clear focal point.

## Distort

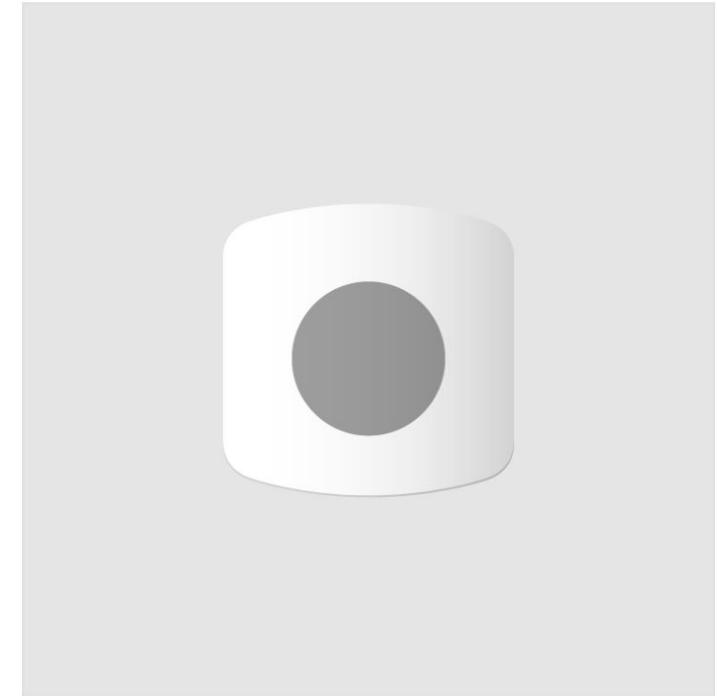


Elements should remain in their geometric form and not be skewed, rotated, bowed, warped, or bent.



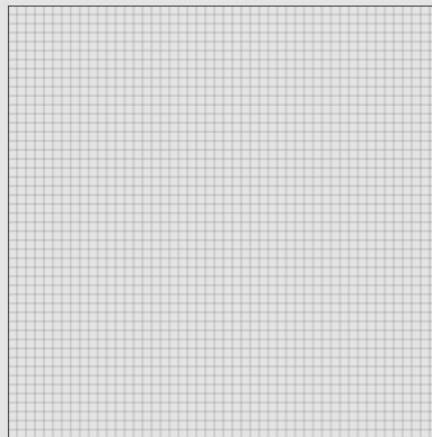
Don't

Don't distort or transform product icons.

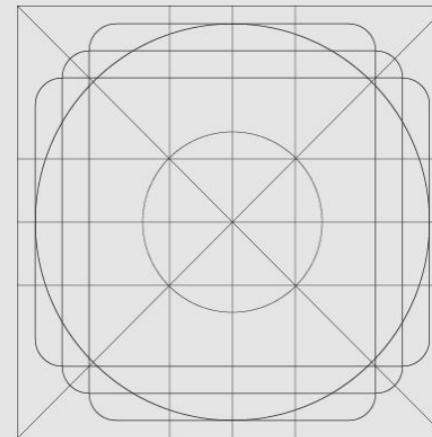


Don't

The icon grid establishes clear rules for the consistent, but flexible, positioning of graphic elements.



Grid

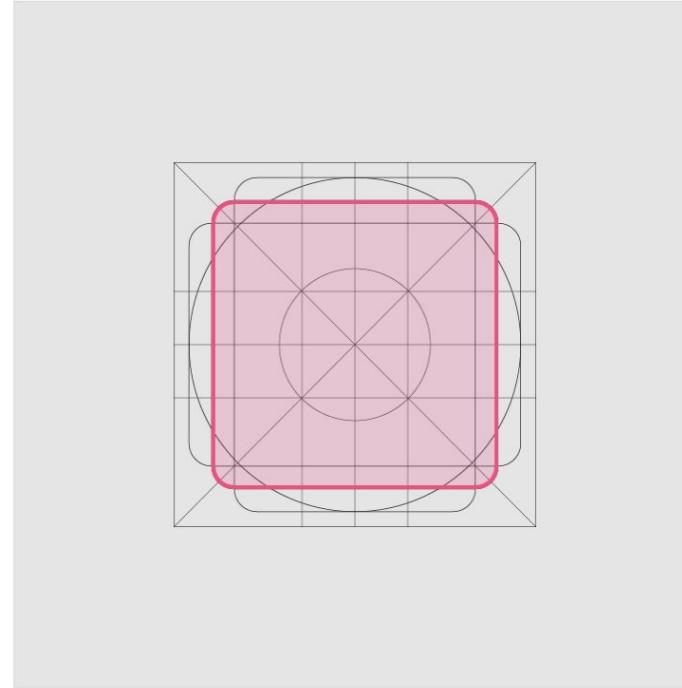


Keyline

# Keyline Shapes

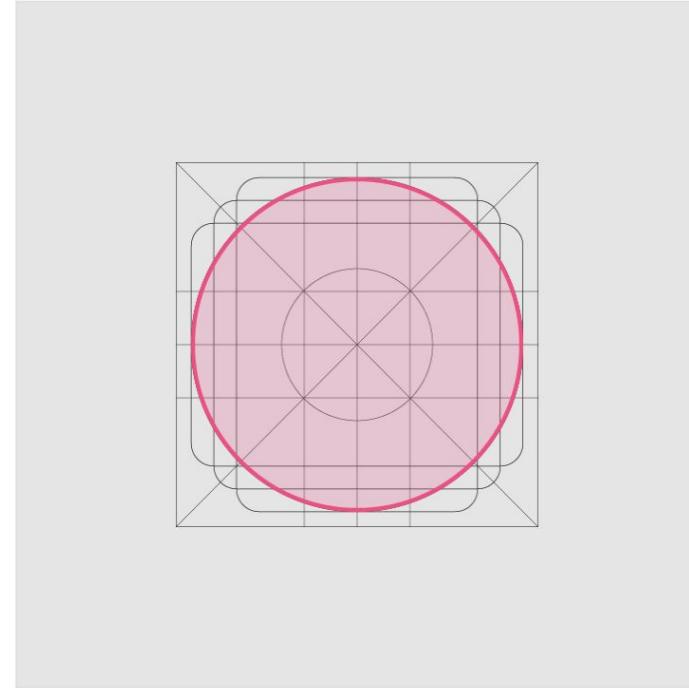


Keyline shapes are based on the grid. By using these core shapes as a baseline, you can maintain consistent visual proportions throughout your product icons.



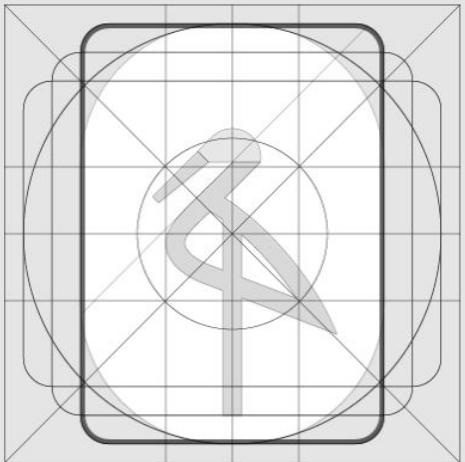
## Square

Height: 152dp  
Width: 152dp

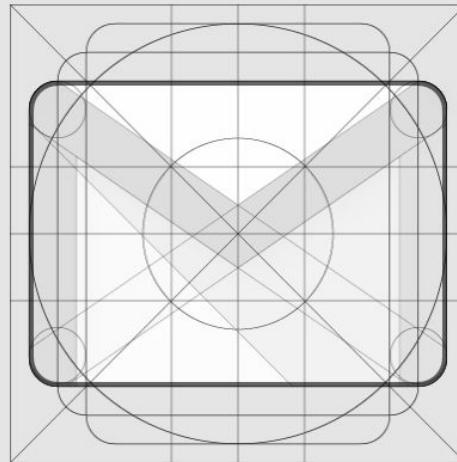


## Circle

Diameter: 176dp



Vertical rectangle



Horizontal rectangle

# Ikoner Resurs

[source: developer.android.com](#)

[source: material.io](#)



## Problem

Kravspecifikation för Android  
Ikoner?



## Solution

Gör aldrig ikonen för avancerad.  
Fyll ut till kanter, avrunda ej, lägg  
inte till för mycket detaljer eller  
skuggning. Följ tipsen!



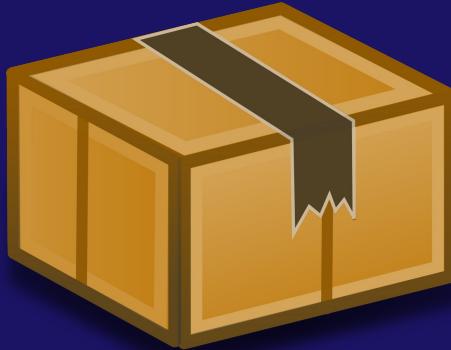
*Frågor?*



# Good Package Naming

# Package Naming

domain.companyName.appName

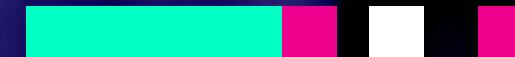


com.krillinator.lektion1

## Choose a good package name

Make sure you choose a package name that is suitable over the life of your application. You cannot change the package name after you distribute your application to users. You can set the package name in application's manifest file. For more information, see the [package](#) attribute documentation.

# Logging



# Logging



## Turn off logging and debugging

Make sure you deactivate logging and disable the debugging option before you build your application for release. You can deactivate logging by removing calls to `Log` methods in your source files. You can disable debugging by removing the `android:debuggable` attribute from the `<application>` tag in your manifest file, or by setting the `android:debuggable` attribute to `false` in your manifest file. Also, remove any log files or static test files that were created in your project.

Also, you should remove all `Debug` tracing calls that you added to your code, such as `startMethodTracing()` and `stopMethodTracing()` method calls.

**!** **Important:** Ensure that you disable debugging for your app if using `WebView` to display paid for content or if using JavaScript interfaces, since debugging allows users to inject scripts and extract content using Chrome DevTools. To disable debugging, use the `WebView.setWebContentsDebuggingEnabled()` method.

# Cleaning Project



# Cleaning Project



## Clean up your project directories

Clean up your project and make sure it conforms to the directory structure described in [Android Projects](#). Leaving stray or orphaned files in your project can prevent your application from compiling and cause your application to behave unpredictably. At a minimum you should do the following cleanup tasks:

- Review the contents of your `jni/`, `lib/`, and `src/` directories. The `jni/` directory should contain only source files associated with the [Android NDK](#), such as `.c`, `.cpp`, `.h`, and `.mk` files. The `lib/` directory should contain only third-party library files or private library files, including prebuilt shared and static libraries (for example, `.so` files). The `src/` directory should contain only the source files for your application (`.java` and `.aidl` files). The `src/` directory should not contain any `.jar` files.
- Check your project for private or proprietary data files that your application does not use and remove them. For example, look in your project's `res/` directory for old drawable files, layout files, and values files that you are no longer using and delete them.
- Check your `lib/` directory for test libraries and remove them if they are no longer being used by your application.
- Review the contents of your `assets/` directory and your `res/raw/` directory for raw asset files and static files that you need to update or remove prior to release.

# Manifest.xml



# Manifest.xml

Review and update your manifest and Gradle build settings

Verify that the following manifest and build files items are set correctly:



- `<uses-permission>` element

You should specify only those permissions that are relevant and required for your application.

- `android:icon` and `android:label` attributes

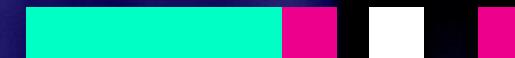
You must specify values for these attributes, which are located in the `<application>` element.

- `android:versionCode` and `android:versionName` attributes.

We recommend that you specify values for these attributes, which are located in the `<manifest>` element. For more information see [Versioning your Application](#).

There are several additional manifest or build file elements that you can set if you are releasing your application on Google Play. For example, the `android:minSdkVersion` and `android:targetSdkVersion` attributes, which are located in the `<uses-sdk>` element. For more information about these and other Google Play settings, see [Filters on Google Play](#).

# Cryptographic Keys



# Cryptographic Keys

## Gather materials and resources

To prepare your app for release, you need to gather several supporting items. At a minimum, this includes cryptographic keys for signing your app and an app icon. You might also want to include an end-user license agreement.

### Cryptographic keys

Android requires that all APKs are digitally signed with a certificate before they are installed on a device or updated. For [Google Play Store](#), all apps created after August 2021 are required to use [Play App Signing](#). But uploading your AAB to Play Console still requires you to sign it with your developer certificate. Older apps can still self-sign, but whether you're using Play App Signing or you're self-signing, you must sign your app before you can upload it.

To learn about certificate requirements, see [Sign your app](#).



**Important:** Your app must be signed with a cryptographic key that has a validity period ending after October 22, 2033.

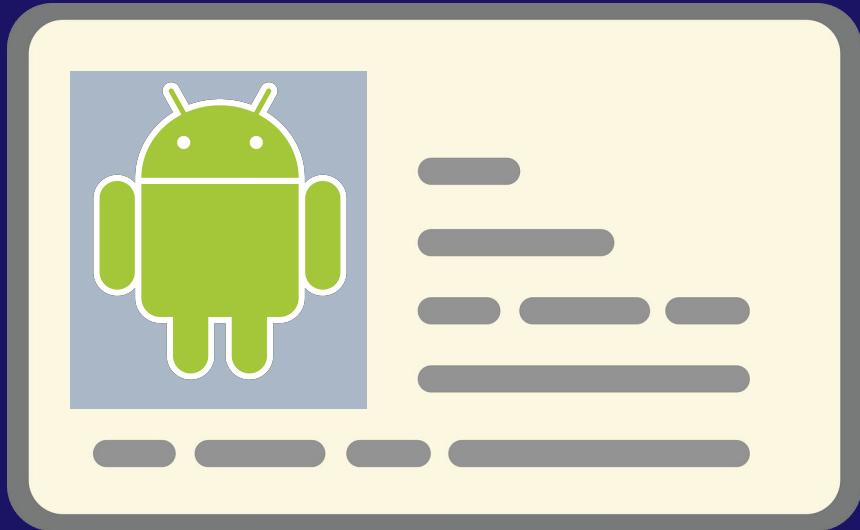
You might also have to obtain other release keys if your app accesses a service or uses a third-party library that requires you to use a key that is based on your private key.

<https://developer.android.com/studio/publish/preparing>

# Preparing for Release



# Preparing for Release



## Implement licensing (if you are releasing on Google Play)

If you are releasing a paid application through Google Play, consider adding support for Google Play Licensing. Licensing lets you control access to your application based on whether the current user has purchased it. Using Google Play Licensing is optional even if you are releasing your app through Google Play.

## Address compatibility issues

Android provides several tools and techniques to make your application compatible with a wide range of devices. To make your application available to the largest number of users, consider doing the following:

- **Add support for multiple screen configurations**

Make sure you meet the [best practices for supporting multiple screens](#). By supporting multiple screen configurations you can create an application that functions properly and looks good on any of the screen sizes supported by Android.

- **Optimize your application for Android tablet devices.**

If your application is designed for devices older than Android 3.0, make it compatible with Android 3.0 devices by following the guidelines and best practices described in [Optimizing Apps for Android 3.0](#).

- **Consider using the Support Library**

If your application is designed for devices running Android 3.x, make your application compatible with older versions of Android by adding the [Support Library](#) to your application project. The Support Library provides static support libraries that you can add to your Android application, which enables you to use APIs that are either not available on older platform versions or use utility APIs that are not part of the framework APIs.

# Preparing for Release

<https://developer.android.com/studio/build/build-variants>

## Building Your application for release

After you finish configuring your application you can build it into a release-ready APK file that is signed and optimized. The JDK includes the tools for signing the APK file (Keytool and Jarsigner); the Android SDK includes the tools for compiling and optimizing the APK file. If you are using Android Studio or you are using the Gradle build system from the command line, you can automate the entire build process. For more information about configuring Gradle builds, see



## Problem

Äldre enheter?



## Solution

Arbeta med androidX paketen för bakåtkompatibilitet.  
Appen ska se bra ut oberoende av skärmstorlekar!



*Frågor?*



# 03

## Preparing for Release

# APK & AAB



# APK & AAB



APK

*Android Package Kit*

APK är den äldre varianten!

“Android kräver att alla ‘APKs’ ska signeras med ett certifikat innan de installeras på en enhet eller uppdateras.

För appar som distribuerar med **APK-filer** i **Play Butik** (skapat före augusti 2021) eller i andra butiker måste du signera dina **APK-filer manuellt för uppladdning**”

# APK & AAB



AAB

*Android App Bundle*

“När du släpper med Android **AAB-bundle**,  
måste du signera ditt **AAB-bundle** med en **uppladdnings nyckel** innan du laddar  
upp det till **Play Console**, och **Play App Signing** tar hand om resten.”

# APK & AAB

Från och med nu så är det ett **MÅSTE** att publicera via AAB

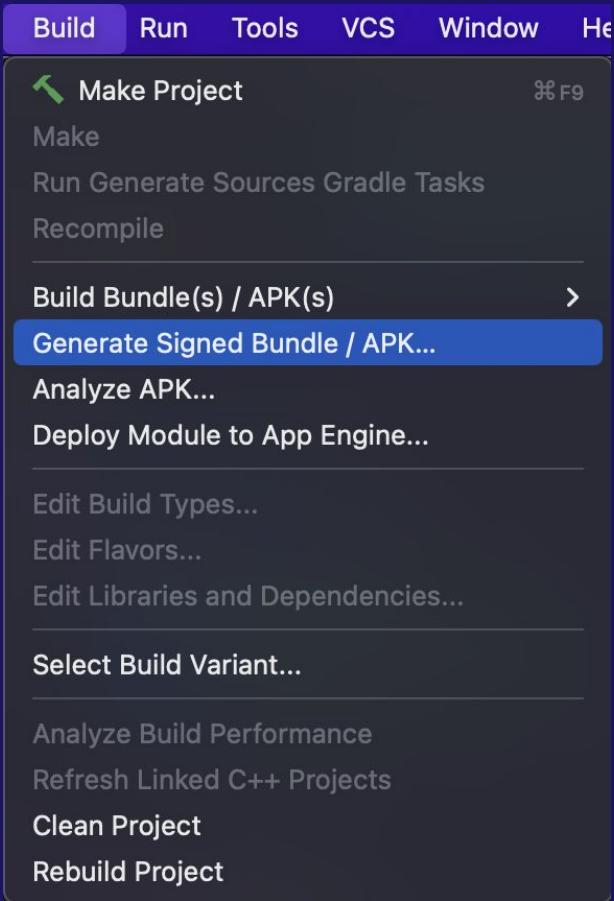
★ **Important:** From August 2021, new apps are required to publish with the [Android App Bundle](#) on Google Play. New apps larger than 150 MB are now supported by either [Play Feature Delivery](#) or [Play Asset Delivery](#).

An *Android App Bundle* is a publishing format that includes all your app's compiled code and resources, and defers APK generation and signing to Google Play.

Google Play uses your app bundle to generate and serve optimized APKs for each device configuration, so only the code and resources that are needed for a specific device are downloaded to run your app. You no longer have to build, sign, and manage multiple APKs to optimize support for different devices, and users get smaller, more-optimized downloads.

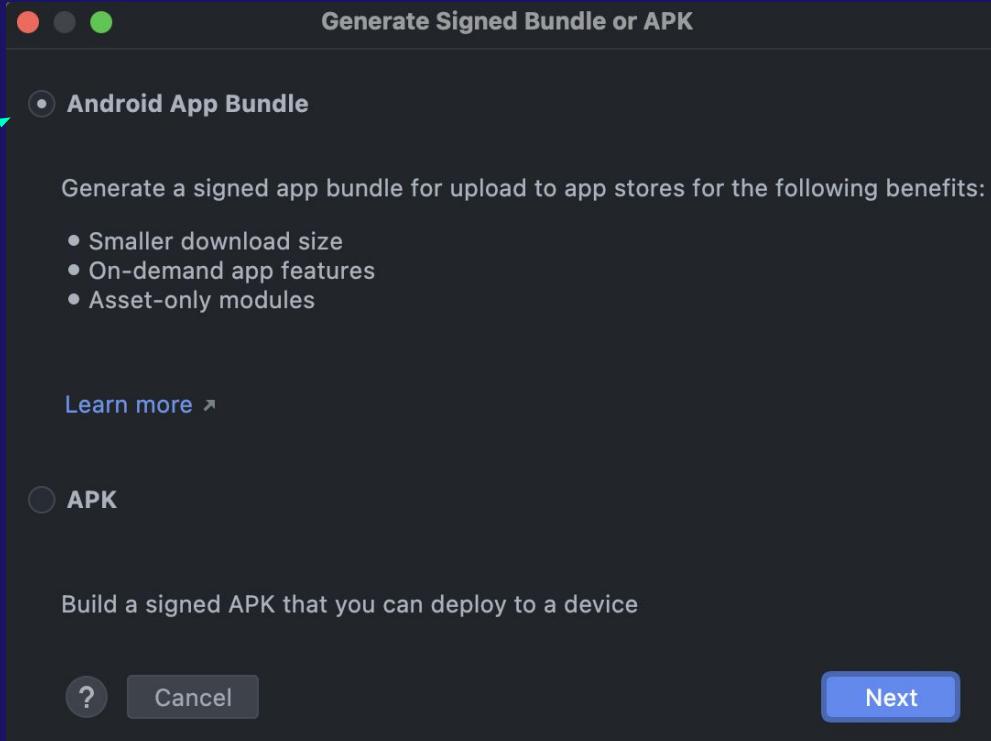
# Step #1

Gå till 'BUILD' fliken, tryck på:  
**'Generate Signed BUndle / APK'**



# Step #2

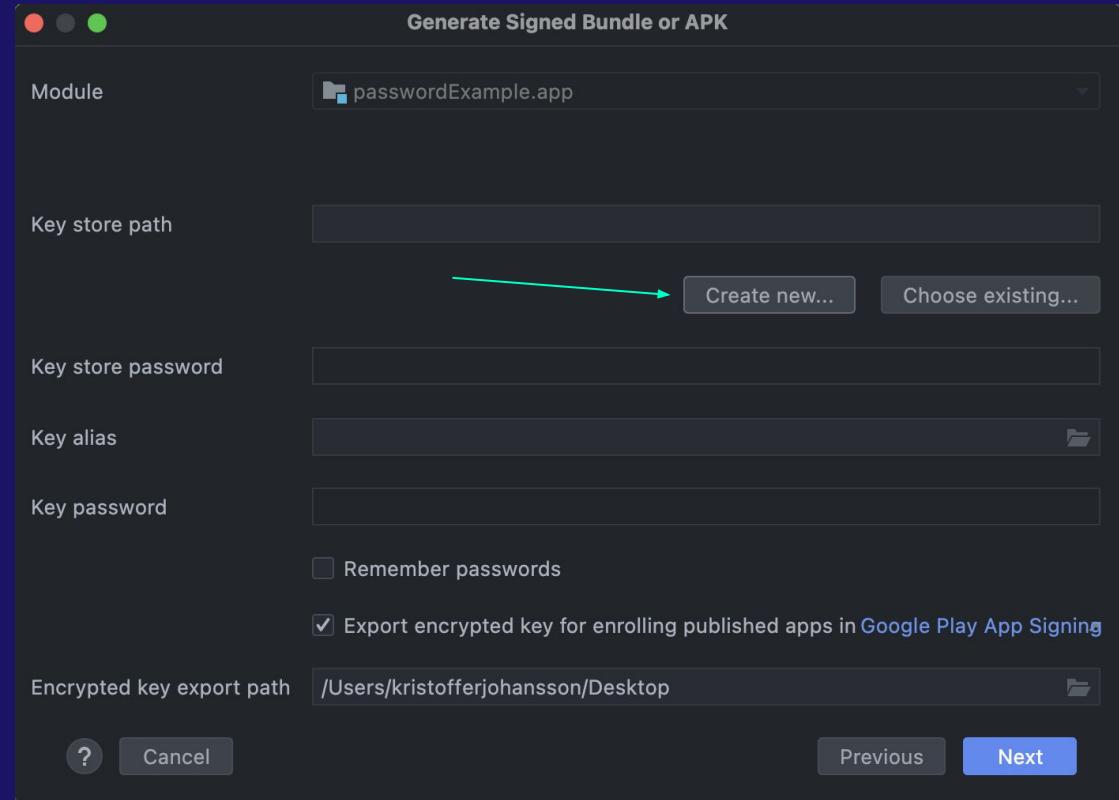
Välj Android App Bundle



# Step #3

Vi behöver skapa en 'key store',  
Om ni inte redan har detta:

Tryck på '**Create New**'



# Step #4

Fyll i formuläret!

New Key Store

Key store path: /Users/krillinator/Desktop/myDemo5

Password:  Confirm:

Key

Alias: keyDemo5

Password:  Confirm:

Validity (years): 25

Certificate

First and Last Name: Benny björnsson

Organizational Unit: Mobile

Organization: Benny incorporated

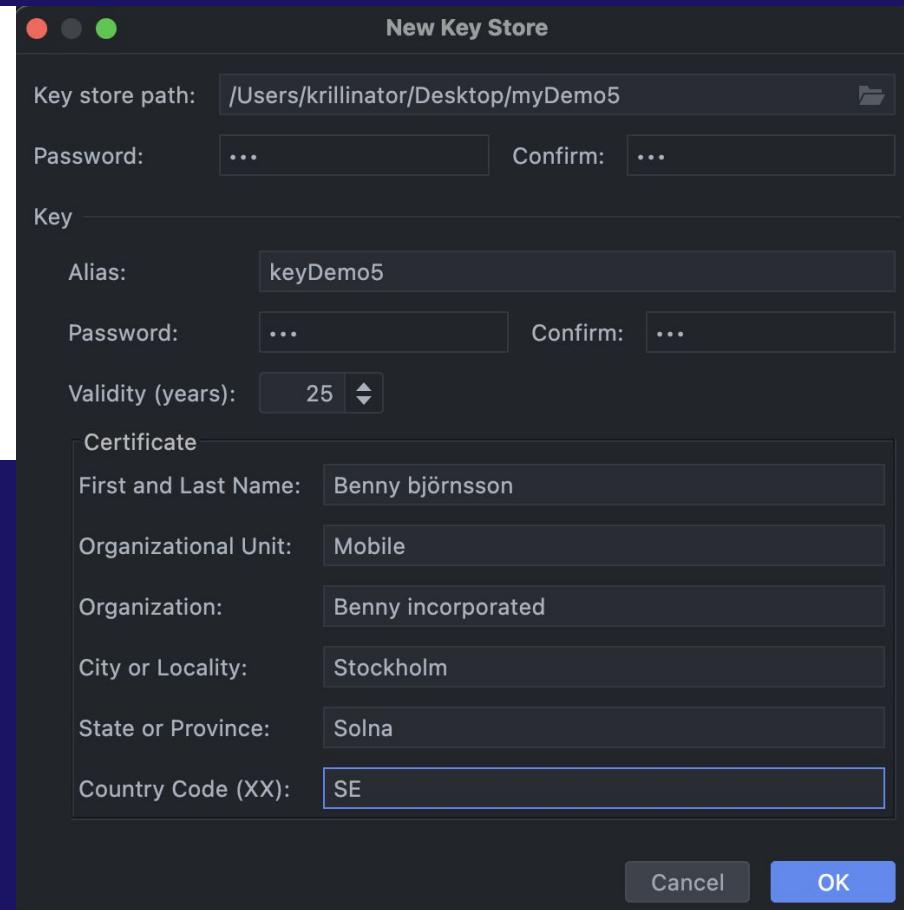
City or Locality: Stockholm

State or Province: Solna

Country Code (XX): SE

## 5. Keystore

- **Key store path:** Select the location where your keystore should be created. Also, a file name should be added to the end of the location path with the `.jks` extension.
- **Password:** Create and confirm a secure password for your keystore.

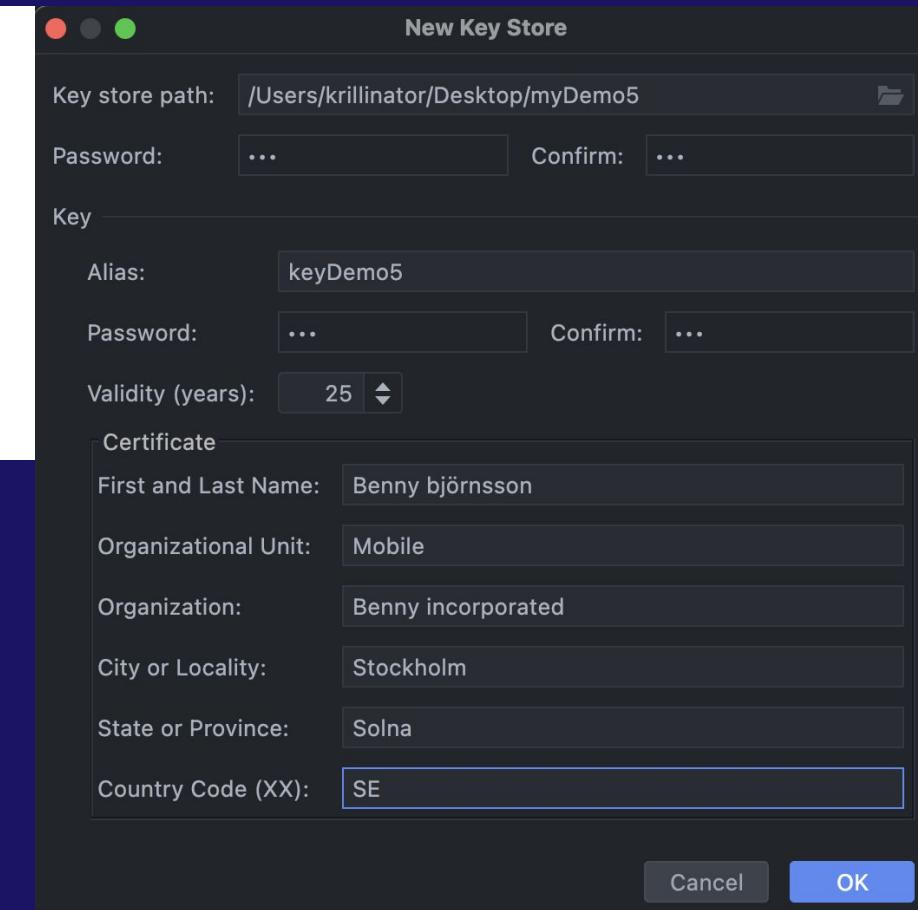


## 5. Keystore

- **Key store path:** Select the location where your keystore should be created. Also, a file name should be added to the end of the location path with the `.jks` extension.
- **Password:** Create and confirm a secure password for your keystore.

## 6. Key

- **Alias:** Enter an identifying name for your key.
- **Password:** Create and confirm a secure password for your key. This should be the same as your keystore password. (Please refer to the [known issue](#) for more information)
- **Validity (years):** Set the length of time in years that your key will be valid. Your key should be valid for at least 25 years, so you can sign app updates with the same key through the lifespan of your app.

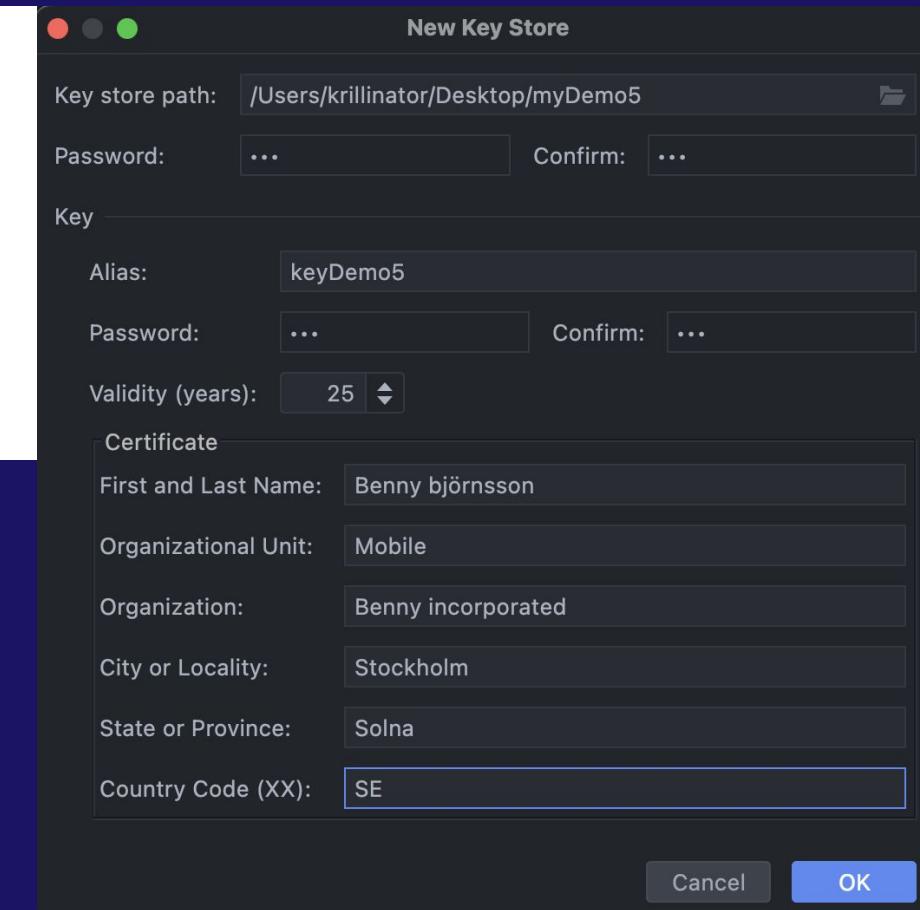


## 5. Keystore

- **Key store path:** Select the location where your keystore should be created. Also, a file name should be added to the end of the location path with the `.jks` extension.
- **Password:** Create and confirm a secure password for your keystore.

## 6. Key

- **Alias:** Enter an identifying name for your key.
- **Password:** Create and confirm a secure password for your key. This should be the same as your keystore password. (Please refer to the [known issue](#) for more information)
- **Validity (years):** Set the length of time in years that your key will be valid. Your key should be valid for at least 25 years, so you can sign app updates with the same key through the lifespan of your app.



# Halfway done!

Nu är du redo för nästa steg!

Om du hellre vill så kan du gå in hit istället:

[https://developer.android.com/studio/publish/app-signin  
g#sign\\_release](https://developer.android.com/studio/publish/app-signing#sign_release)

# Step #5

5. If you're signing an app bundle with an existing app signing key, and you'd like to later opt your app in to Play App Signing, check the box next to **Export encrypted key** and specify a path to save your signing key as an encrypted `*.pepk` file. You can then use your encrypted app signing key to [opt in an existing app into Play App Signing](#).
6. Click **Next**.
7. In the next window (shown in figure 4), select a destination folder for your signed app, select the build type, choose the product flavor(s) if applicable.
8. If you are building and signing an APK, you need to select which **Signature Versions** you want your app to support. To learn more, read about [app signing schemes](#)

# Step #5

Generate Signed Bundle or APK

Module

Key store path

Key store password

Key alias

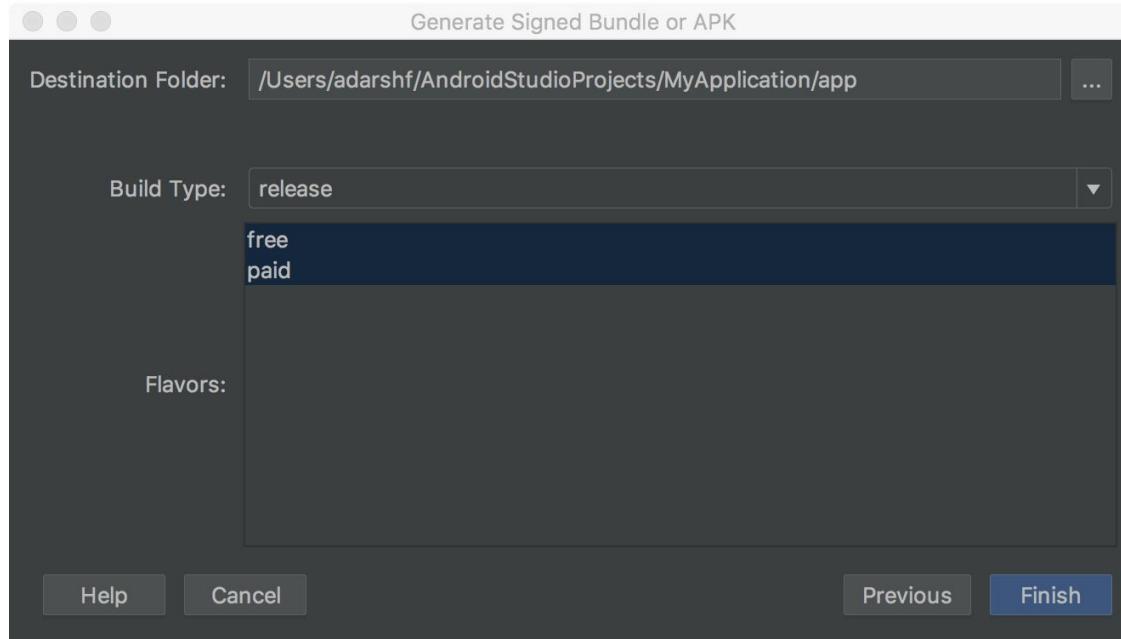
Key password

Remember passwords

Export encrypted key for enrolling published apps in Google Play App Signing

# Step #6

9. Click **Finish**.

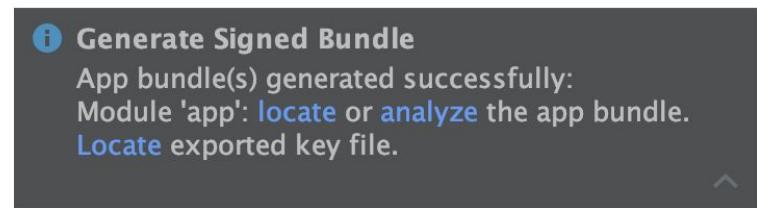


**Figure 4.** Generate a signed version of your app for the selected product flavors.

# Step #7

After Android Studio finishes building your signed app, you can either **locate** or **analyze** your app by clicking on the appropriate option in the pop-up notification. If you selected the option to export your signing key, you can quickly navigate to it by clicking the dropdown arrow in the bottom right corner of the popup to expand it and clicking **Show Exported Key File**, as shown in figure 5.

Now you're ready to opt your app in to Play App Signing and upload your app for release. If you're new to the app publishing process, you may want to read the [Launch overview](#). Otherwise, continue to the page about how to [Upload your app to the Play Console](#).



**Figure 5.** Click the link in the popup to analyze or locate your app bundle, or locate your exported signing key.

# Using Play App Signing



Google Play Console

play.google.com/console

Google Play Console

All applications

Dashboard

Inbox

Statistics

Managed publishing

Release

Grow

Quality

Monetize

Policy and compliance

Policy status

App content

Search

Shrine

App content

Let us know about the content of your app. This is to make sure your app complies with Google Play policies. [Learn more](#)

To do

Privacy policy

Not started • Add a privacy policy to your store listing

Adding a privacy policy to your store listing helps provide transparency about how you treat sensitive user and device data

Start

Ads

Not started • Let us know whether your app contains ads

You must let us know whether your app contains ads. The 'Contains ads' label is shown next to apps with ads on Google Play. Make sure this information is accurate, and is kept up to date.

Start

App access

Not started • Provide instructions on how to access restricted parts of your app

If parts of your app are restricted based on login credentials, memberships, location, or other forms of authentication, provide instructions on how to access them

Start

Google Play Console,  
är en HUB för era  
applikationer där ni kan se  
allt som är relevant till just  
er applikation!



*Om du inte redan har ett konto så kan du skapa  
ett här: <https://play.google.com/console/>*

# Skapa Konto



Slutför köpet

Developer Registration Fee

25,00 US\$

Betalningsuppgifterna sparas i Google-kontot. Hantera dina betalningsmetoder på [pay.google.com](https://pay.google.com).



Lägg till ett kredit- eller bankkort



# Skapa en Version

Step by step!

<https://developer.android.com/studio/publish/app-signing#enroll>



# Skapa en Version

*Step 1 is signing your app (which is complete)*

2. Sign in to your [Play Console](#) and navigate to your app.
3. On the left menu, click **Release > Setup > App integrity**.
4. If applicable, review the Terms of Service and select **Accept**.
5. Select one of the options that best describes the signing key you want to upload to Google Play and follow the instructions that are shown. For example, if you used Android Studio to export your app's signing key, as described on this page, select **Upload a key exported from Android Studio** and upload the `*.pepk` file for your key.
6. Click **Enroll**.

You should now see a page with the details of your app's signing and upload certificates. Google Play now signs your app with your existing key when deploying it to users. However, one of the most important benefits to Play App Signing is the ability to separate the key you use to sign the artifact you upload to Google Play from the key that Google Play uses to sign your app for distribution to users. So, consider following the steps in the next section to generate and register a separate upload key.

# Skapa en Version

## Steg 1: Skapa en version

En version är en kombination av en eller flera appversioner som du förbereder inför lanseringen eller uppdateringen av en app. Du kan skapa en version via tre olika testkanaler eller i produktionskanalen:

- **Öppet test:** Öppna testversioner är tillgängliga för testare på Google Play. Användare kan gå med i testet via dina butiksuppgifter.
- **Slutet test:** Slutna testversioner är tillgängliga för ett begränsat antal testare som väljs ut av dig. De kan testa förhandsversionen av appen och skicka feedback.
- **Internt test:** Interna testversioner är tillgängliga för upp till 100 testare som väljs ut av dig.
- **Produktionskanal:** Produktionsversioner är tillgängliga för alla Google Play-användare i de utvalda länderna.

Förbereda och lansera

[https://support.google.com/googleplay/android-developer/answer/9859348?visit\\_id=637816528480667213-3370916032&rd=1](https://support.google.com/googleplay/android-developer/answer/9859348?visit_id=637816528480667213-3370916032&rd=1)

# 04

## Uppgifter

&

## Eget Arbete

# Uppgifter

## Välkommen till första uppgiften!

Uppgifterna är till för att testa dina färdigheter och kunskaper för att både öva och repetera på det vi har arbetat med under föreläsningarna.

Dessa är **INTE** obligatoriska.  
Men är starkt rekommenderat att arbeta med.



# MINNS DU?

```
// Vad är 'Google Play' ?  
// Hur urskiljer sig 'console' från  
just 'Google Play'?  
  
// Vad är ett APK?  
// Hur urskiljer sig ett APK mot AAB?
```

```
1           // -Uppgift #1- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Skapa ett nytt projekt!
```

```
6
```

```
7     Döp projektet till: Lektion_6_uppgifter
```

```
8
```

```
9
```

```
10    Generera ett 'Signed Bundle / APK'
```

```
11    Välj 'AAB'
```

```
12 */
```

```
13
```

```
14 // HINT & Examples
```

```
15 hint(" Navigera till BUILD ")
```

```
16
```

```
17
```

```
18
```

```
19
```

```
20
```

```
21
```

```
22
```

```
23
```



*Kom igång enkelt med uppgift #1*

```
1 // -Uppgift #2- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Skapa ett nytt Key Store och fyll
```

```
6     i formuläret!
```

```
7
```

```
8     Glöm inte sedan när detta är skapat att
```

```
9     komma ihåg lösenordet!
```

```
10
```

```
11    Välj sedan att bygga applikationen
```

```
12    som 'debug'!
```

```
13 */
```

```
14
```

```
15 // HINT & Examples
```

```
16 hint(" Slide #51")
```

```
17
```

```
18
```

```
19
```

```
20
```

```
21
```

```
22
```

```
23
```



Enkla men viktiga steg!

```
1           // -Uppgift #3- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Logga in på ditt konto inom Play Console
```

```
6
```

```
7     På vänster meny, tryck på:
```

```
8     Release > Setup > App Integrity
```

```
9
```

```
10    Review Terms of Service - Accept
```

```
11
```

```
12    Tryck på:
```

```
13    Upload a key exported from Android Studio
```

```
14    och ladda upp din fil
```

```
15
```

```
16    Tryck på 'Enroll'
```

```
17
```

```
18 */
```

```
19
```

```
20 WARNING - DETTA STEG KRÄVER ATT DU HAR BETALAT
```

```
21 GOOGLE PLAY $25 USD FÖR SKAPANDET AV DITT KONTO
```

```
22
```

```
23 Denna uppgift är alltså INTE ett krav
```



Step by step, tråkiga steg men som behöver utföras för att äntligen lägga upp sin applikation på Google Play.

# THANKS !

Do you have any questions?  
[kristoffer.johansson@sti.se](mailto:kristoffer.johansson@sti.se)

[sti.learning.nu/](http://sti.learning.nu/)

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.

*You can also contact me VIA Teams (quicker response)  
Du kan också kontakta mig VIA Teams (Snabbare svar)*