

Deploy to Google Play Store

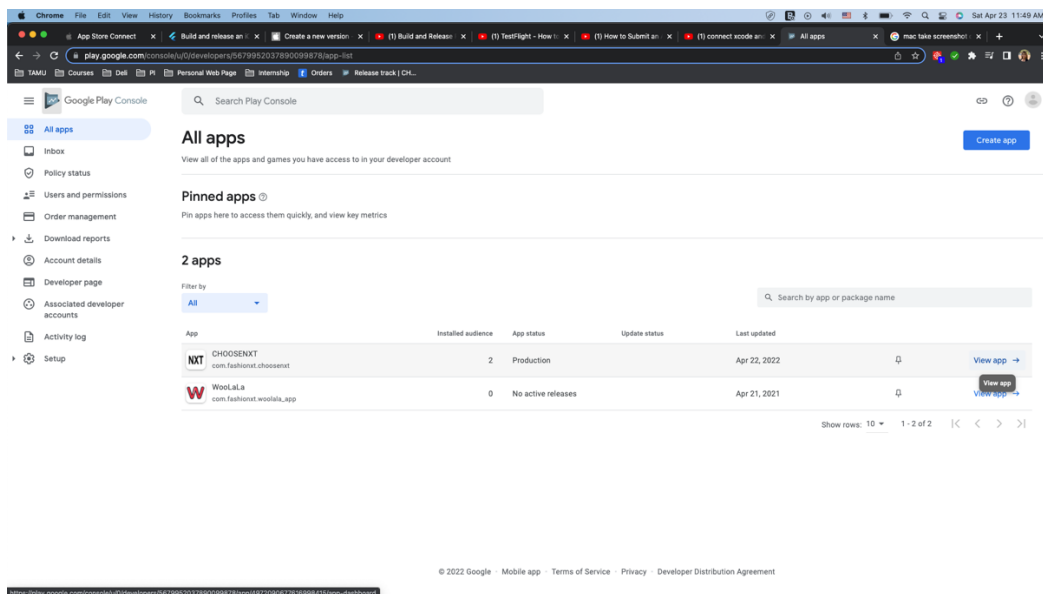
1. Navigate to woolala_app folder
2. Run <flutter clean> to make sure you will not get any errors from the previous builds.
3. Run <flutter pub get> to make sure you have all the required dependencies.
4. Open “woolala_app/pubspec.yaml”
 - Go to line 18. You will see the version here in the format A.B.C+D
 - Modify the version here. Chose A, B, and C according to the amount and importance of the modifications to the app. You can navigate to --- to see the last release’s version.
 - D shows the count for the versions you have released so far. Again, you can navigate to Google Play Console to see the count and to see how many builds have been released so far. If you currently have X number of build, just do $D = X + 1$
5. Get the file “key.jks” from Tito. Place that file inside the “woolala_app/android/app” folder.
6. Create a “key.properties” file inside the “woolala_app/android” folder. Put the following code inside “key.properties”.

```
1 storePassword=  
2 keyPassword=  
3 keyAlias=key  
4 storeFile=/Users/. . ./woolala_app/android/app/key.jks
```

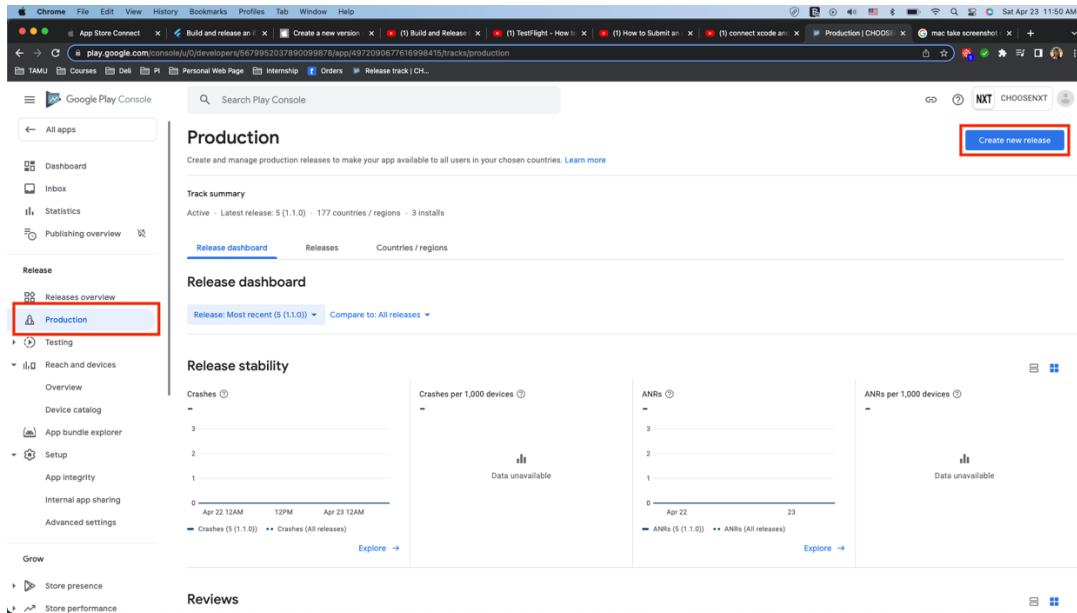
get “storePassword” and “keyPassword” from Tito.

7. Open “woolala_app/android/app/build.gradle”

- Check line 52 and make sure it reads: <targetSdkVersion 30>. You might need to change this if in future you get a related error from ---- while trying to upload your build.
 - Check line 53 and make sure you change this to the same version count you are using in the “pubspec.yaml” file as described in step 4.
 - Check line 54 and make sure you change this to the same version number you are using in the “pubspec.yaml” file as described in step 4.
8. Navigate to woolala_app folder and run <flutter build appbundle>.
 9. The previous step will create the app-release.aab file which you will upload to the Google Play Console. You can find this .abb file under “woolala_app/build/app/outputs/bundle/release” folder.
 10. Open Google Play Console in your browser. Click on view app.



11. Click on the production tap from the bottom left list of items.
12. Click on Create new release.



13. You need to drag and drop the app-release.aab file mentioned in step 9 here.
14. After uploading is done, your building is a working build if you do not get any errors.
Otherwise, you will get an error with a description of what is wrong with the build.
15. Scroll through the page and fill any necessary field according to the details of your release.
16. In the field where you are required to put a description/note about the app, you have the option to copy from the previous releases.
17. Click save.
18. Click review release
19. Click start rollout to production
20. The app will be reviewed by the Google Play Store and if no issues are found it will be released to the store shortly afterwards.