Using iOS Simulator vs Real Device

You might not have a real iOS device, but would still like to use the app on iOS on a simulator. I have a little hack. It’s not pretty, but it works.

1. In Android Studio, navigate to your project and then composeApp/src/iosMain/com.jetbrains.cameraapp/camera. Open the file CameraScreen.ios.kt.
2. Around line 80-81 there should be two lines with one commented out.

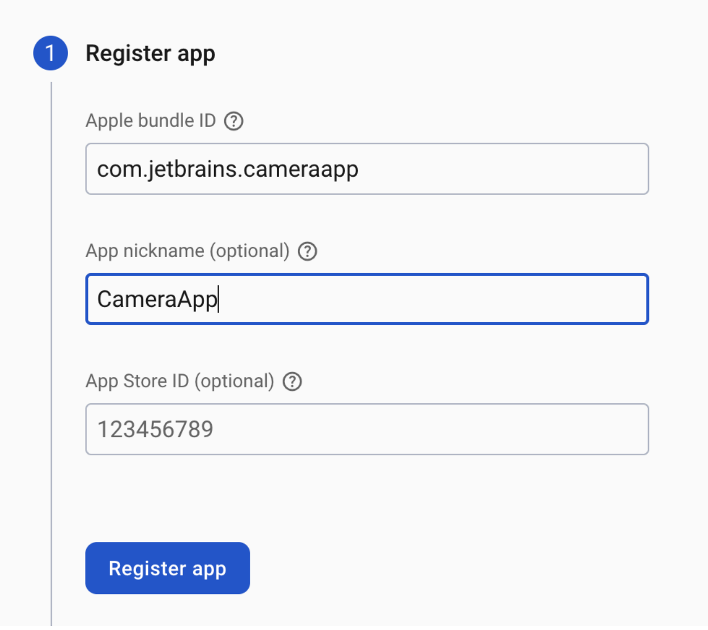
```
SimulatorScreenContents(onPicture, modifier)
// CameraScreenContents(onPicture, modifier)
```

   a. If you want to use the simulator, make sure the CameraScreenContents is commented out, and SimulatorScreenContents in.
   b. If you want to use the real device, make sure the SimulatorScreenContents is commented out, and CameraScreenContents in.

Setting up Firebase for Notifications [optional]

1. Go to https://console.firebase.google.com/ (you need to be logged into a Google account).
2. Click on Create a Firebase project.
3. Call the app “CameraApp”, click on Continue.
4. You will be asked if you want to get help from Gemini. For this project, you don’t need it so slide the slider to off, click on Continue.
5. You also don’t need Analytics, so slide the slider to off, click on Create Project.
6. Wait a little while Firebase sets up your project, then click on Continue.
7. Click on the iOS button.

8. In the Add Firebase to your Apple app screen, fill in the form like so:



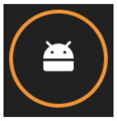
1 Register app

Apple bundle ID: com.jetbrains.cameraapp

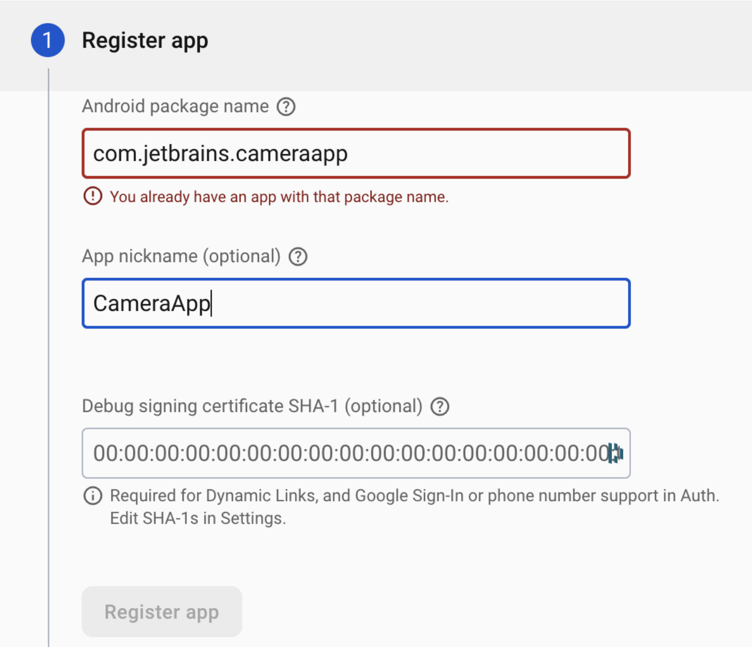
App nickname (optional): CameraApp

App Store ID (optional): 123456789

9. Download the GoogleService-Info.plist file by clicking on the button.
10. Back in Android Studio, add this file to your iOS application in the iosApp/iosApp directory directly below the project. Make sure that the file has the correct name, in case you've downloaded similar files before and now have a (1) or (2) added.
11. Run the iOS app to check everything works including notifications. Remember to switch the code to simulator / real device depending on your setup as in the section above.
12. Navigate back to the project overview page and click on the Android button.



13. In the resulting screen, fill in the form like so (don't worry about the error in this screenshot)



1 Register app

Android package name

com.jetbrains.cameraapp

You already have an app with that package name.

App nickname (optional)

CameraApp

Debug signing certificate SHA-1 (optional)

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

14. Download the google-services.json file by clicking on the button.
15. Back in Android Studio, add this file to your Android app in the composeApp directory. Make sure the file is named exactly like that.
16. Run the Android app as usual and test that it works (including notifications.)