In this lesson we're going to do the final piece of setup to use Firebase and that's adding the Firebase　packages to our project.

After all, the reason Firebase is easy to use is because we're going to be using their classes and their　code as a starting point.

I'll be showing you how to do this for both Android and iOS　in this lesson. The latter part of this lesson is going to include some iOS specific instructions　so just skip the last part of this lesson　if you're developing on windows.

Now once we've added our iOS and Android apps to Firebase, then it's time to actually incorporate the Flutter　packages that will allow us to work with Firebase using Dart.

As I mentioned before, there's a whole bunch of Firebase services that we could be using. But in our case　were only using two of these which is authentication and cloud\_firestore.

So if you click on these links, then it'll take you to the relevant flflutter package on the dartlang　pub page. And whenever you're working with anything from Firebase, you'll always need to start by adding　firebase\_core.

It's the one that all of these other packages are going to depend on.

So let's first install firebase\_core into our pubspec. So let's go into Installing and I'm just gonna　copy this dependency as it is and I'm going to be adding it to the pubspec.yaml file.

So just below are animated text kit　I'm gonna add the firebase\_core and then we'll add the firebase\_auth.

Now at the time when you're looking at this, these numbers might have changed but just take whatever　it is that you see there and you can paste it into our pubspec folder just so that you get the latest　versions.

All right.

So now that we're done adding all of our dependencies, we're ready to go ahead and hit up packages get　to incorporate all of those things into our project. Once that's done,　it's really important to test our app on iOS　if you're testing on iOS and also Android. And it's important to test it from a cold start.

So we're going to click stop to stop all the processes that are happening which means that all our apps　will exit on Android and on iOS. And now we're going to start with Android and we're going to click Run　to build our app from fresh.

So after the usual deprecation warnings that we're getting here, we should be getting our APK which　is our Android file being built.

And if we take a look at our emulator, we should be looking at a working version of our app as it is.

So this is success and we're now ready to test the iOS linking up.

So it's really important that you test things at this stage because later on as we add more code, different　things can go wrong.

And if you leave it too late, it might just not work.

So once you're done with the Android testing, go ahead and choose your iOS simulator or device. And before　you hit run, one of the things I recommend you to do is to go into this tab in Android Studio called　the terminal. And here we're going to update CocoaPods.

Now in the very very beginning of the course, we installed CocoaPods but it's been many many moons ago　and there might have been a newer version out.

So it's really important that after you add those Firebase packages to your pubspec file that you first　do an update on CocoaPods before you run it for iOS otherwise terrible things might happen and we might　have to figure out how to fix it, if that is the case.

Prevention is better than cure.

So first things first, we're going to update our pod repo so we're going to put pod repo update and this　grabs the latest versions for our CocoaPods.

And that might take a little while but it should be done after a minute or two.

Now the thing I didn't mention is that I'm going to be cutting out and speeding up many part of this　because it can take a while especially updating CocoaPods.

It is something that can be a little bit time consuming but hey all good things are worth waiting for　right? Now once it's done you should see your prompt, so that dollar sign, and the name of your user.

So this is now done. And now we're ready to move onto the next step of updating CocoaPods which is going　to be sudo gem install cocoapods.

And this is going to take even longer　and if you're using sudo which means super user do, we're going to have to enter the admin password,so the password that you use to log onto your Mac to access your computer. Now because I've already updated　very very recently, it's not doing a lot of work and it finished after about five seconds.

But this can take anywhere, I've seen in the past students, take anywhere between five seconds and five　hours.

So it really is something that can take a while.

You just have to be patient, it hasn't crashed.

Just let it go on until it's done.

And once it's done, you'll see your user name and the dollar sign or the prompt again and we're ready　to do the final part which is just to run pod setup. And this should just do all the sort of behind the　scenes linking up.

And once that's done　it'll tell you that it's done and it'll say setup completed. Now if at any point during this process　you see some red text in here instead of green or yellow or white, then make sure you dive deeper into　those error messages or google for it or usually they'll give you a hint as to what's going on. But everything　going well there shouldn't be any problems　and we're now ready to shrink down our terminal and go ahead and stop our app and run it on iOS.

So let's go ahead and hit run and let's see what happens.

So now that's complete.

And even though we've got quite a few messages in here coming from the Firebase packages, we've got our　app up and running and it's all going well even if it did take quite a while.

Now notice how we're running pod install and that's Flutter which is initiating that to be able to　install all of these packages as cocoapods to our iOS app.

Now if you encounter any errors at any point be sure to refer to the course resources where I'll provide　a link to some of the most common errors that you'll see when trying to incorporate Firebase to your　Flutter app.

And I've got some good solutions for those things.

But the best way is just to follow this lesson step by step closely.

This is the easiest way to have the smoothest experience and if you do encounter any issues, it might be　worth reviewing the videos and some of those steps that you might have missed out.

But once you've confirmed that your app can run from cold start on iOS as and Android, if you're testing　on both, then you're ready to proceed to the next step where we actually get started registering some　of our users.

So all of that and more, I'll see you on the next lesson.

自動スクロール