[https://firebase.google.com/](https://firebase.google.com/" \t "_blank)  
  
<https://firebase.google.com/docs/ios/setup>  
  
<https://firebase.google.com/docs/ios/installation-methods?authuser=0#cocoapods>  
  
<https://github.com/firebase/firebase-ios-sdk>  
  
In case you wanted to create diagrams, you can use this app, it’s pretty good  
<https://app.diagrams.net/>   
  
can you share with us the library of messageUI  
<https://github.com/MessageKit/MessageKit>  
  
<https://messagekit.github.io/>

<https://firebase.google.com/docs/database/ios/read-and-write>

This link explains how to structure your database from Firebase <https://firebase.google.com/docs/database/ios/structure-data>

**Firebase Authentication**

Firebase Authentication

*What is Firebase Authentication?*

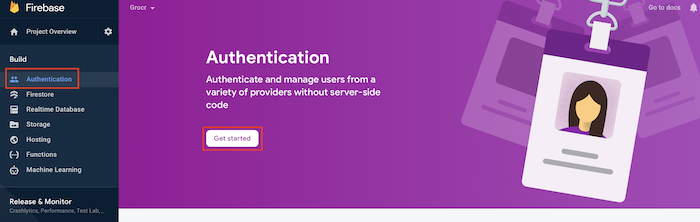
Firebase Authentication is a backend service provided by Google that makes it easy to login users to your app. It supports different types of authentication such as email and password, anonymous login and phone numbers login. It can also authenticate with different providers such as Google, Facebook, and Twitter.

When should we use Firebase Authentication?

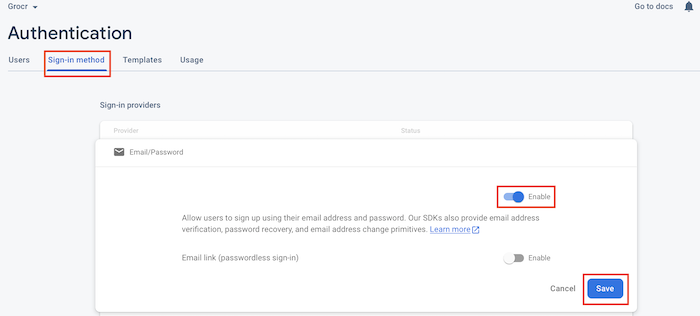
We should use Firebase Authentication anytime the user needs to login or sign up for a mobile application.

Setting up email and password authentication

To enable email and password authentication, go to Firebase Dashboard. Click “ *Authentication”* and then click *“Get Started.”*



Select “ *Sign-in-method.”* In Sign-in-providers, select the *Email/Password* row. Toggle *Enable* and click *Save.*



Now we have authentication set up in our Firebase Dashboard.

Add Firebase Dependencies to XCode project

Lets go to our XCode project and add the dependencies for Firebase Authentication using CocoaPods.

If you are not sure that CocoaPods is install in your computer then open Terminal and type this command to download CocoaPods

sudo gem install cocoapods

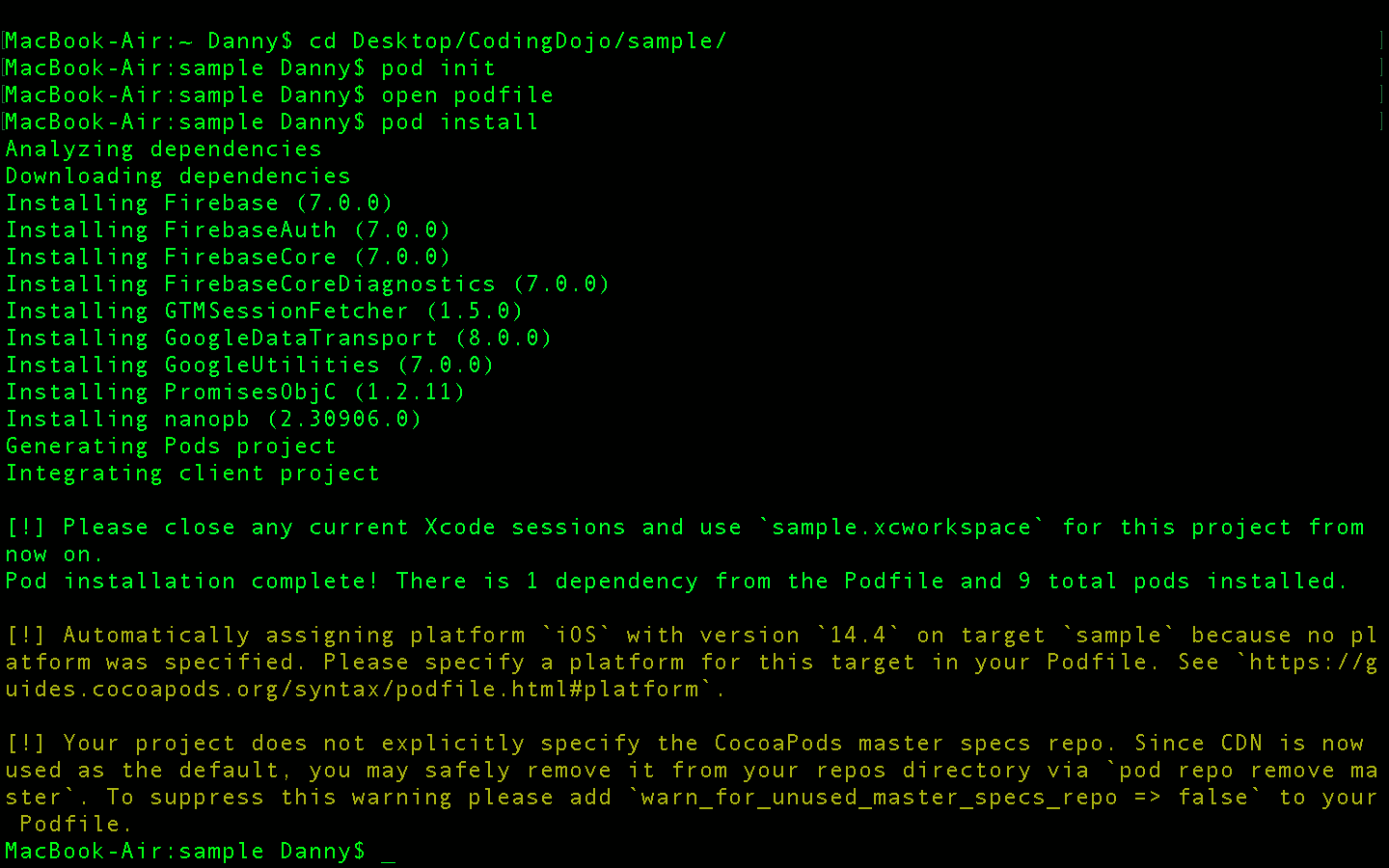
For more information about CocoaPods go to this website

https://guides.cocoapods.org/using/getting-started.html

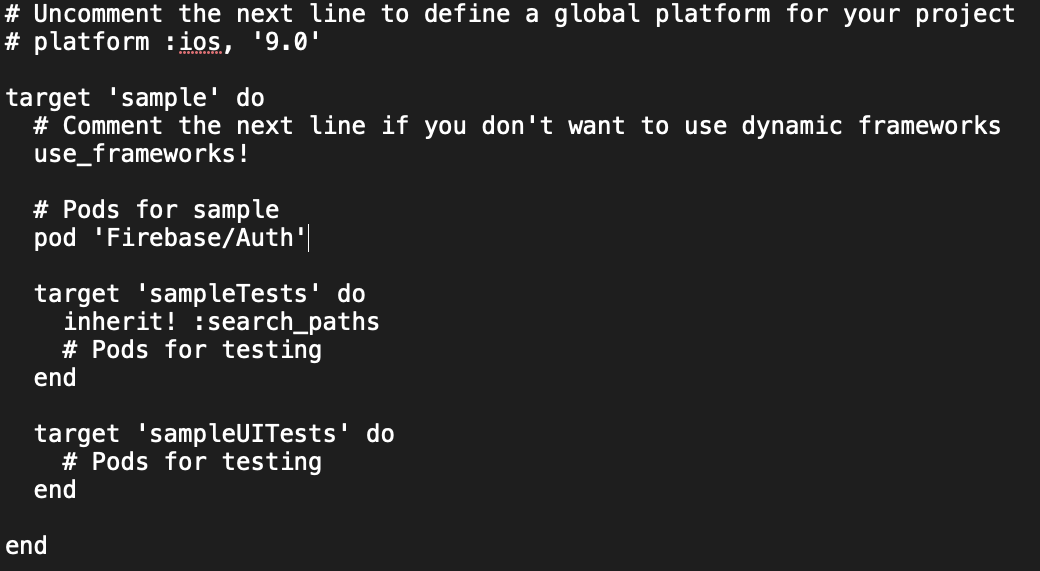
First let's open our Podfile.

* To create a Podfile.
  + Open terminal
  + Navigate to your project location using *“cd”* For example if your project is in a folder called “Projects” in Desktop. Then to navigate to my projects I will type cd Desktop/Projects
  + Type *pod init*
  + Type *open podfile*
  + In your Podfile type *pod 'Firebase/Auth'*
  + Click File -> Save
  + Go back to terminal and type *pod install*

After typing ***pod install.*** You should see all the dependencies being added to your project. From now on we are going to open our. **xcworkspace.** We will no longer use **.xcodeproj.**



The Podfile should look something like this.



In the image above, we are just adding **pod ‘Firebase/Auth’**

Signing up a user

Now we have set up the dependencies using CocoaPods and enabled email and password authentication in our Firebase Dashboard.

We will now go over the code for signing up a user to Firebase

Auth.auth().createUser(withEmail: String, password:String) { (authResult: AuthDataResult?, error: Error?) in

// code ..

}

What is Auth?

* Auth manages Authentication for Firebase apps.

What is auth()?

* auth() gets the auth object for the default Firebase apps.

What is createUser()?

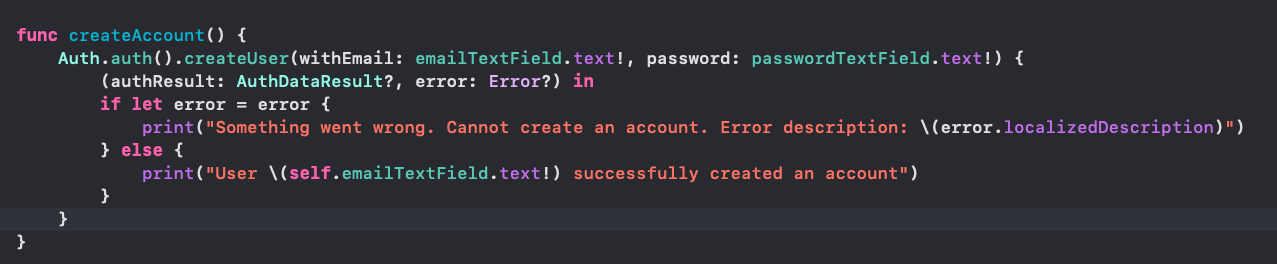
* createUser() creates and, on success, signs in a user with the given email address and password.
* Keep in mind that createUser() takes in 2 parameters: email and password. Both of these data type is a String.

What isAuthDataResult?

* AuthDataResult is a helper object that contains the result of a successful sign-in, link and reauthenticate action.

We will now add code in our *ViewController.swift* file to create an account.

Below I created a function that has the logic to create an account.



Log in a user

We will now go over the code for login in a user to Firebase.

Auth.auth().signIn(withEmail: String, password: String) { (authResult: AuthDataResult?, error: Error?) in

// code ..

}

As you can see the code is very similar to the previous. The only difference is that it has a function called signIn()

What is signIn()?

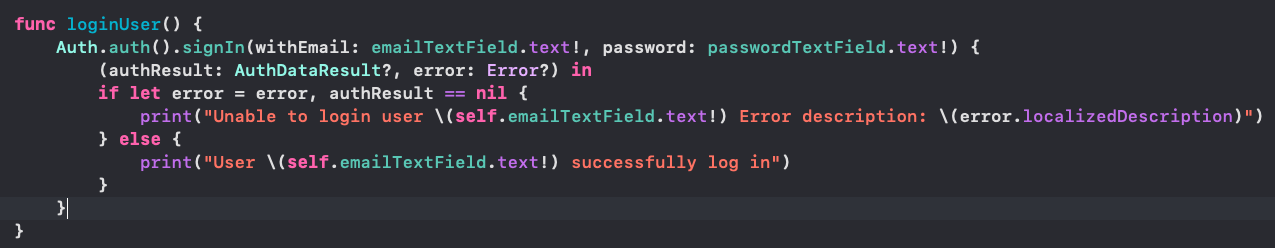
* In this example signIn() sign in a user with an email and password.

What is AuthDataResult?

* AuthDataResult is a helper object that contains the result of a successful sign-in, link and reauthenticate action.

We will now add code in our *ViewController.swift* file to login a user.

Below I created a function that has the logic to login a user.



Authentication State

What is Authentication state?

* Authentication state is an observer where Firebase checks the current user login state
* Authentication state can either have 2 status
  + Current user is logged in.
  + Current user is not logged in.

Why is Authentication state important?

* During the development cycle it is best practice to make the user have a great experience using your mobile application.
* In this case if the user closes your app and tries to re open your app. We should check if the current user is still logged in. If the current user is still logged in we should skip the login page and proceed to the home page.

Lets look over the code for the authentication state

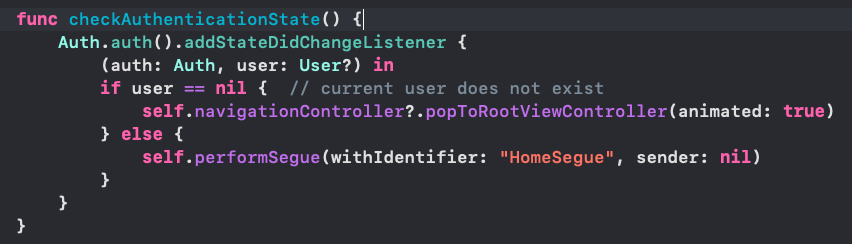
Auth.auth().addStateDidChangeListener() { (auth: Auth, user: User?) in

// code ..

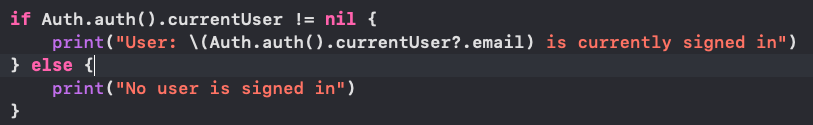
}

What is addStateDidChangeListener()?

* addStateDidChangeListener() is a block that listens to any changes to the authentication state of a user.



You can also get the current user by using the currentUser property.



**Firebase Setup**

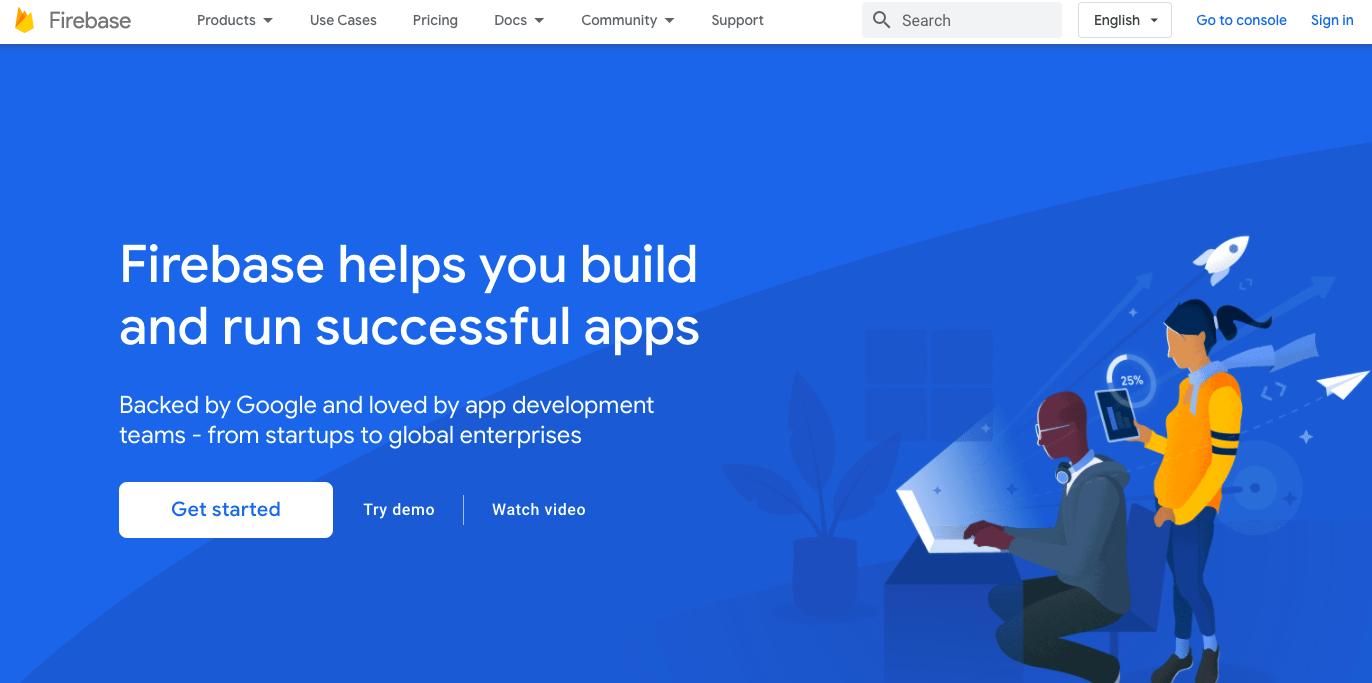
* Setting up Firebase Account

What is Firebase? Firebase is a platform developed by Google for creating web and mobile applications.

What is Firebase used for? Firebase is a backend platform for building iOS, Android, and Web apps.

What features does Firebase provide? Firebase provides tools such as Authentication, Firestore database, Realtime database, Storage, hosting, cloud messaging, Analytics, bug fixing and much more.

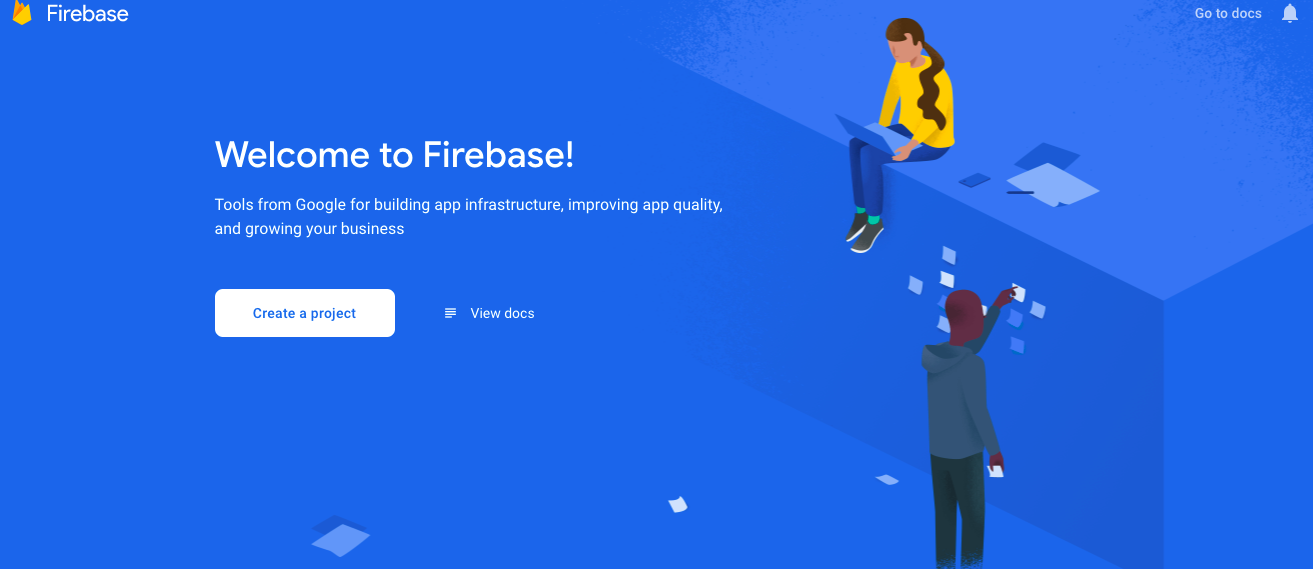
How to get started with Firebase?



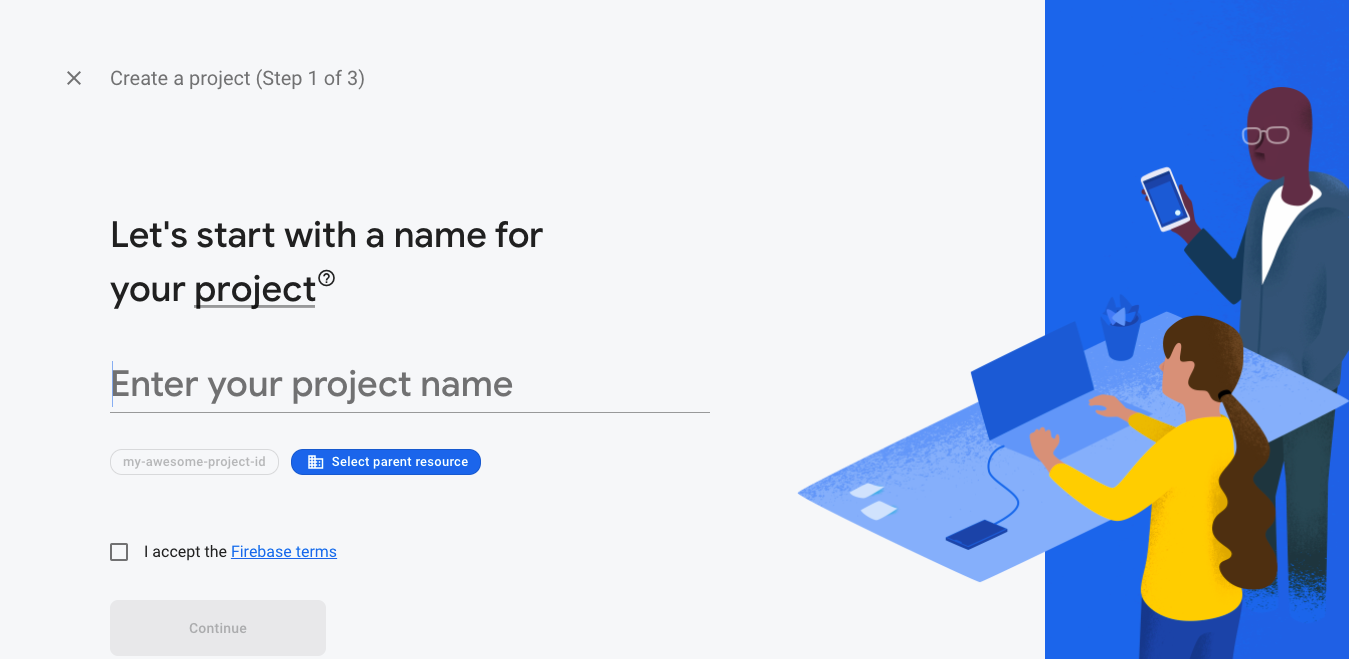
Go to this link.

https://firebase.google.com/

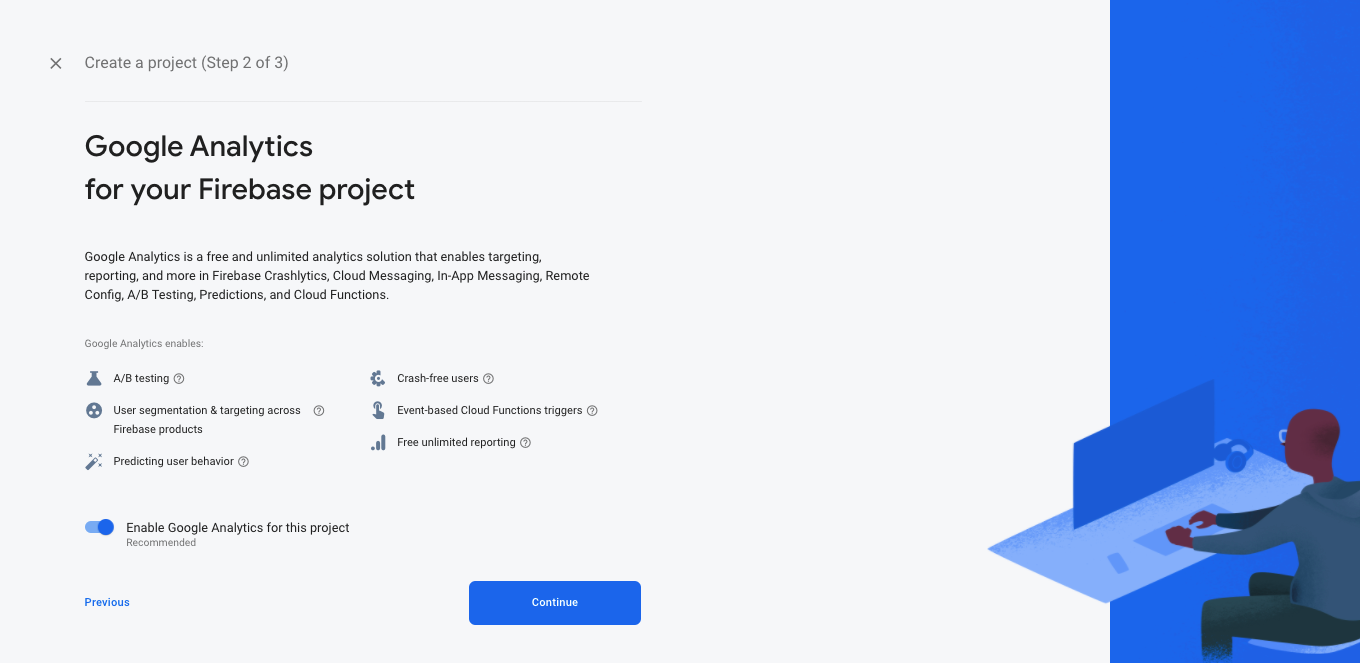
Create an account by clicking on Go to Console on the top right. You will need to login using your Google account.



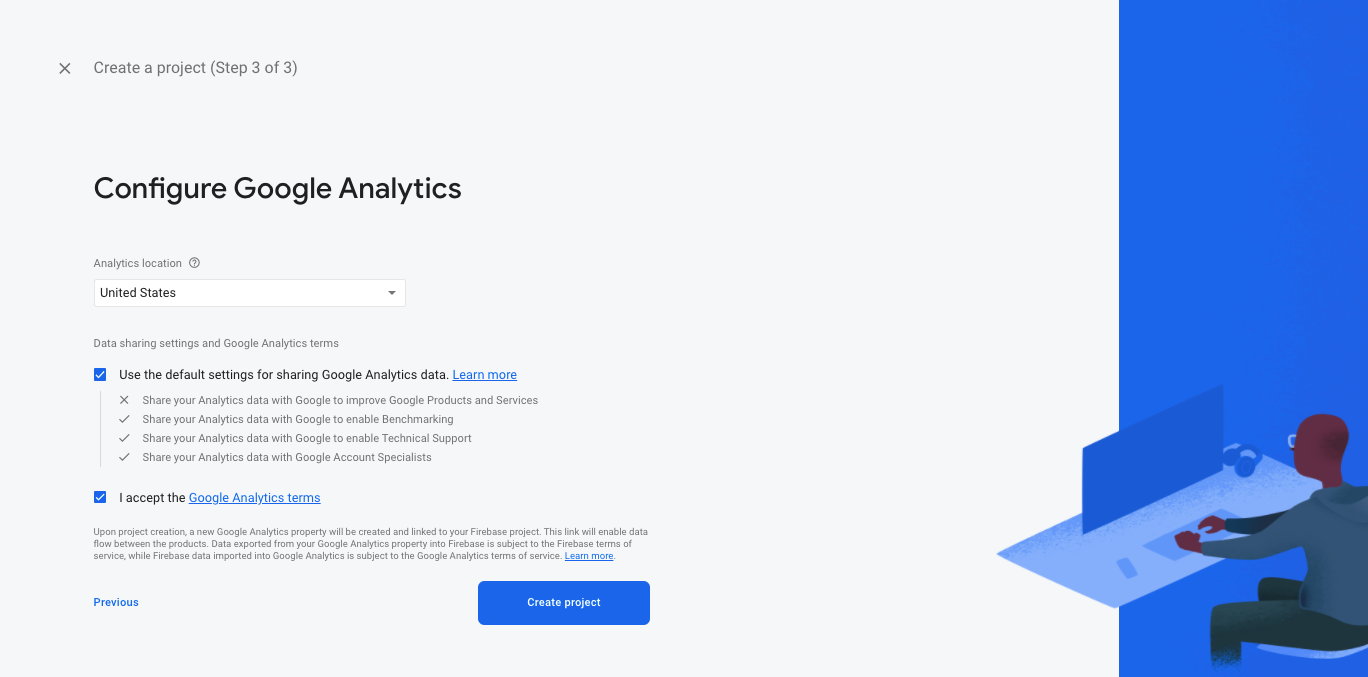
Click on create a project. Enter a project name and accept terms and conditions. Once this is done, click Continue.



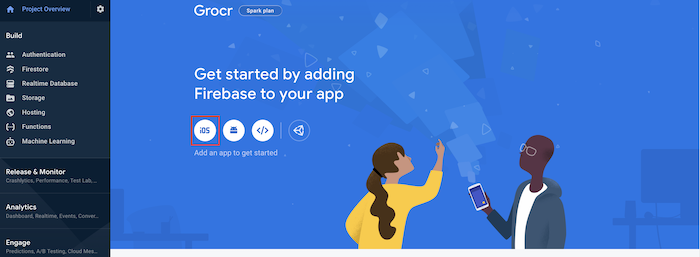
Click Continue.



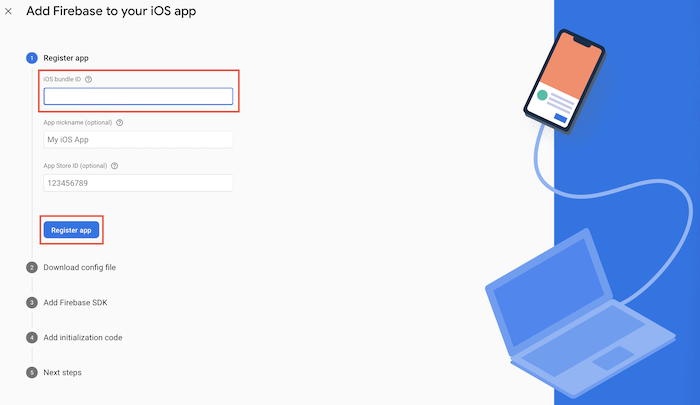
Click on your current location and accept the terms and conditions. Once this is completed click on Create Project.



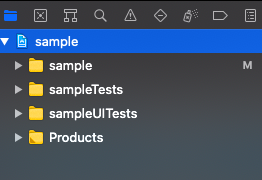
After creating a project click on iOS.



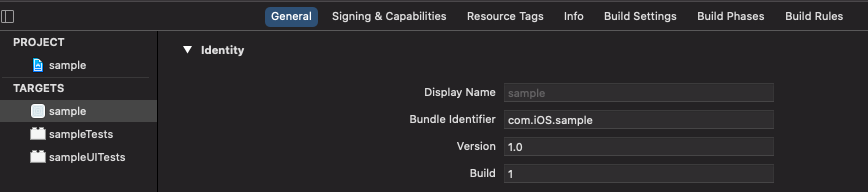
The iOS Bundle ID must be written exactly the same as the Bundle Identifier in XCode.



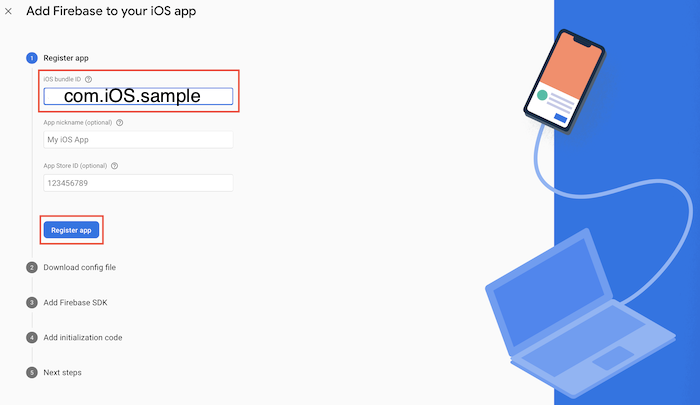
Open your iOS project in XCode, select the top-level app in the project navigator



Select the General tab.



The Bundle Identifier from XCode must be written the same as the iOS Bundle ID in Firebase.



Click Download GoogleService-Info.plist Follow the instructions and move it to the root of your Xcode project.

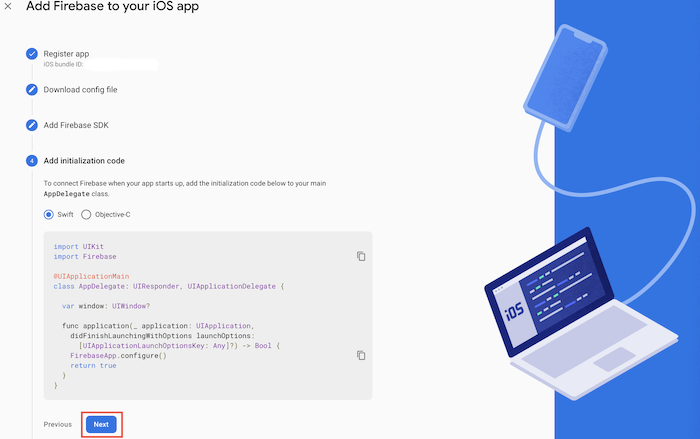


We are going to use CocoaPods to install dependencies. In this case we will be installing Firebase to our XCode project. Open terminal and navigate to the location where your Xcode project is located. For example, if our project is in Desktop then type cd Desktop. Type the code below. This will initialize and install a Podfile.

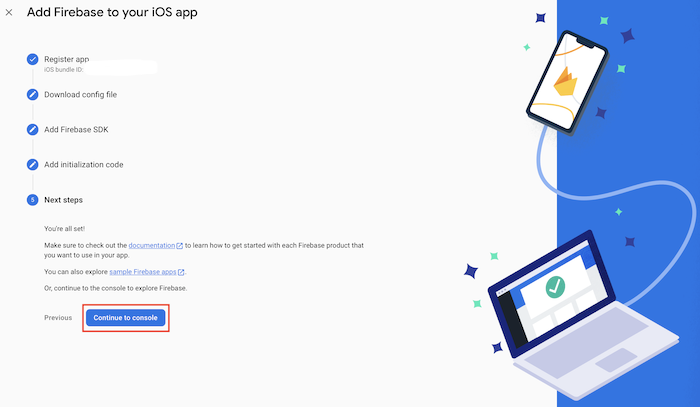


After installing the Podfile. You will see a file with a .xcworkspace. We will be now using this file for our project. We will no longer use the .xcodeproj file.

The sample code below is an example of how to configure and initialize firebase for our app. We will add this code in our AppDelegate file in Xcode. Click next when this step is completed.



Click continue to finish setting up Firebase installation



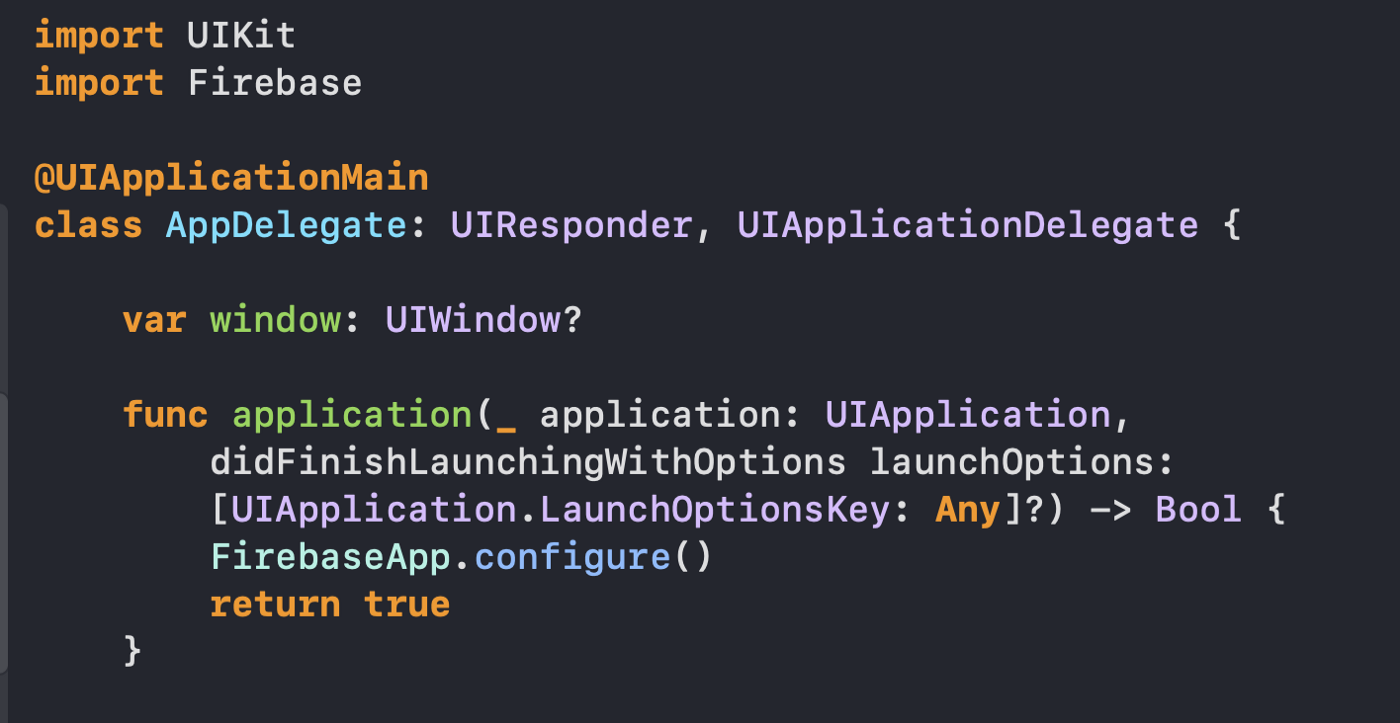
Initializing Firebase to XCode

In XCode, open *AppDelegate.swift.* Add this code on top of your program to import Firebase.

import Firebase

In XCode, open *AppDelegate.swift.* Add this code before the return statement inside the *application(*: didFinishLaunchingWithOptions:)\_

FirebaseApp.configure()



Once everything is setup you should see your project name you created. Click on your project name and you will see the main menu of Firebase Dashboard.

The image above is an example of the Firebase Dashboard. The following section has Authentication, Firestore Database, Realtime Database, Storage, Hosting, Analytics and much more.

