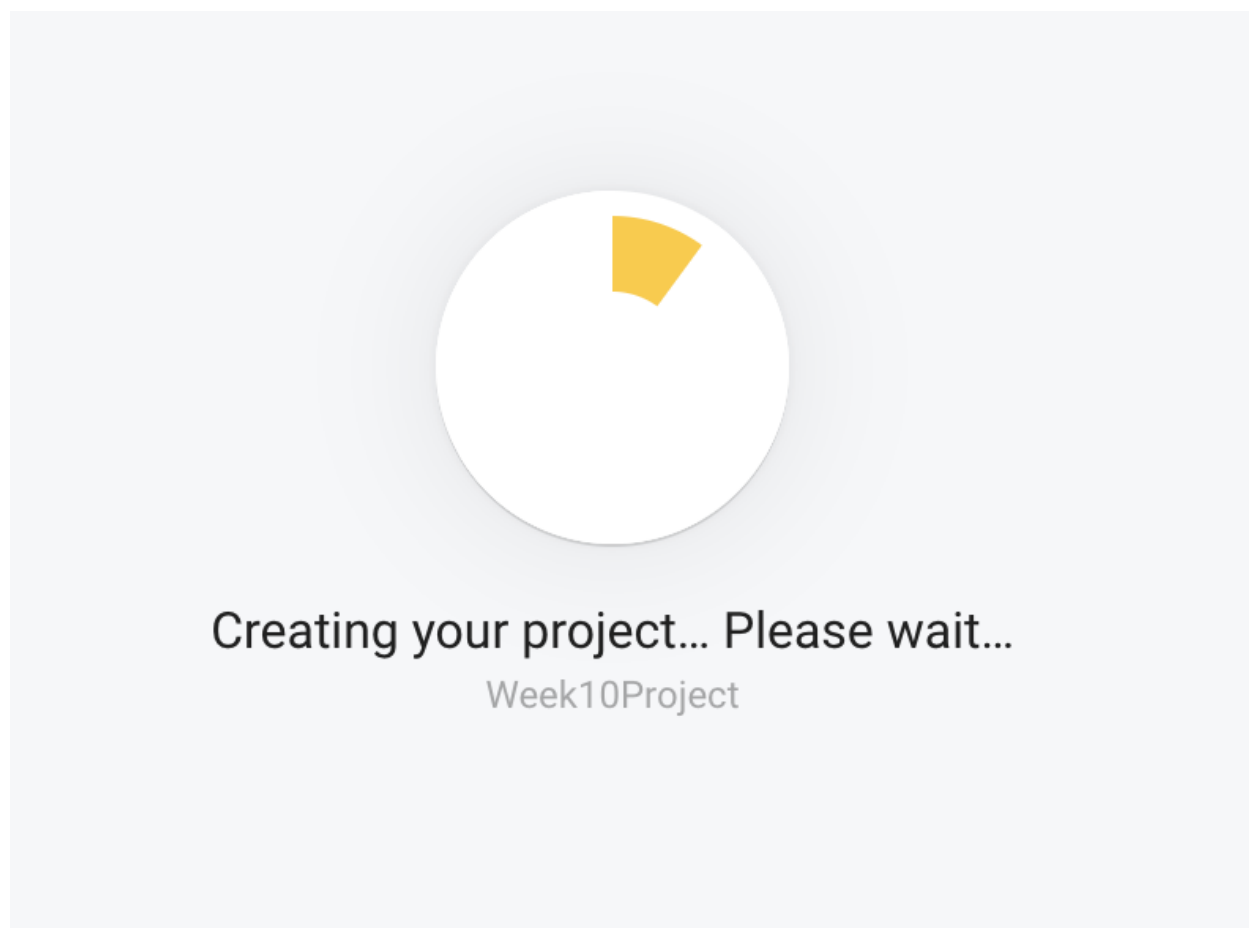


Setup Firestore

In the Firebase console, create a new project.

- Give your project any name and then tap next. Firestore will attempt to create the project.



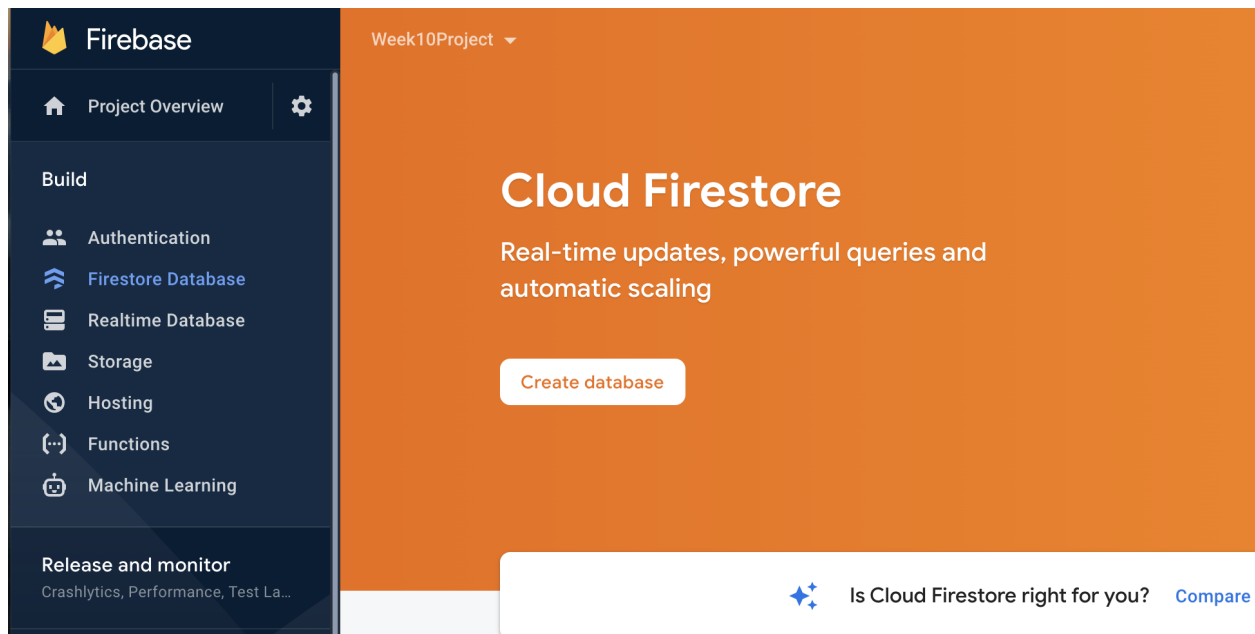
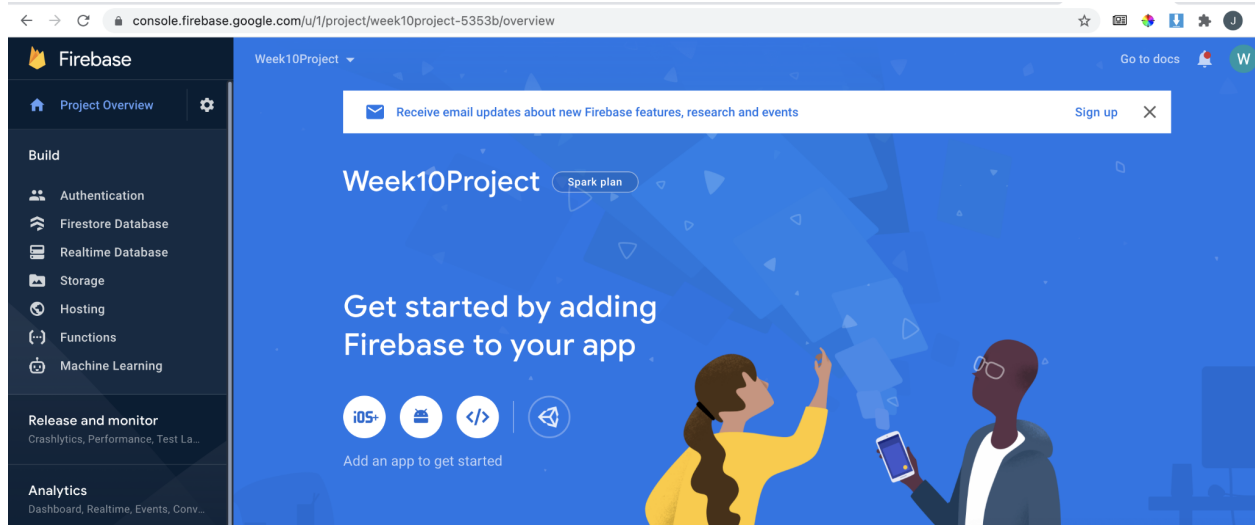


Week10Project



Your new project is ready

Continue



Create database



- 1 Secure rules for Cloud Firestore — 2 Set Cloud Firestore location

After you've defined your data structure, **you will need to write rules to secure your data.**

[Learn more](#)

- ☐ **Start in production mode**
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

- ☒ **Start in test mode**
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2021, 12, 15);
    }
  }
}
```

! The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Cancel

Next

Create database



- ✓ Secure rules for Cloud Firestore — 2 Set Cloud Firestore location

Your location setting is where your Cloud Firestore data will be stored.

! After you've set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

[Learn more](#)

Cloud Firestore location

nam5 (us-central)

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Cancel


Enable

Create database

✓ Secure rules for Cloud Firestore

2 Set Cloud Firestore location

Your location setting is where your Cloud Firestore data will be stored.

 After you've set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

[Learn more](#)

Provisioning Cloud Firestore...

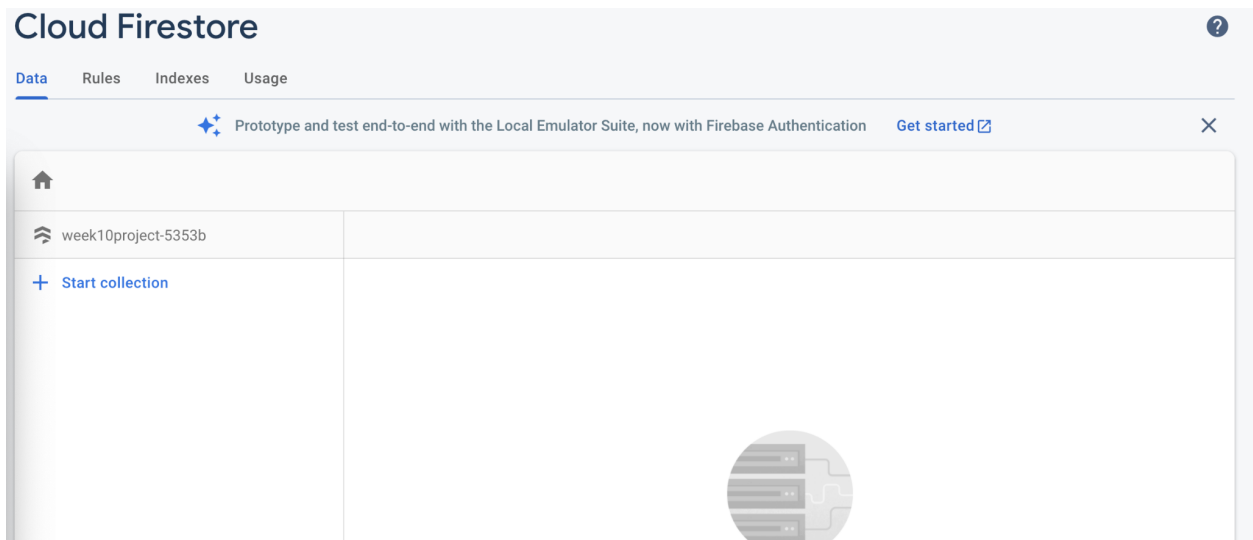
Cloud Firestore location

nam5 (us-central) ▼

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

CancelEnable

When finished:



Start a collection

- 1 Give the collection an ID
- 2 Add its first document

Parent path

/

Collection ID

A collection is a set of documents that contain data

Example: Collection "users" would contain a unique document for each user

Cancel

Next

Use the user interface to define a document for your collection

- For each field, you can choose from a variety of data types

Start a collection

- ✓ Give the collection an ID ——— 2 Add its first document

Document parent path

/movies

Document ID

Auto-ID

Required

Field	Type	Value	
title	= string	Avengers: End Game	—
genre	= string	Action	—
runningTime	= number	150	—
<div>+</div>			

Cancel

Save

Your document must have an id

- The ID can either be manually created (by you), or autogenerated by Firestore
- To have Firestore auto generate an id, click the Auto-ID button

Document ID

Auto-ID

! Required

Start a collection

- ✓ Give the collection an ID — 2 Add its first document

Document parent path

/movies

Document ID

7dIgNEbcKvu7fiqbpXx2

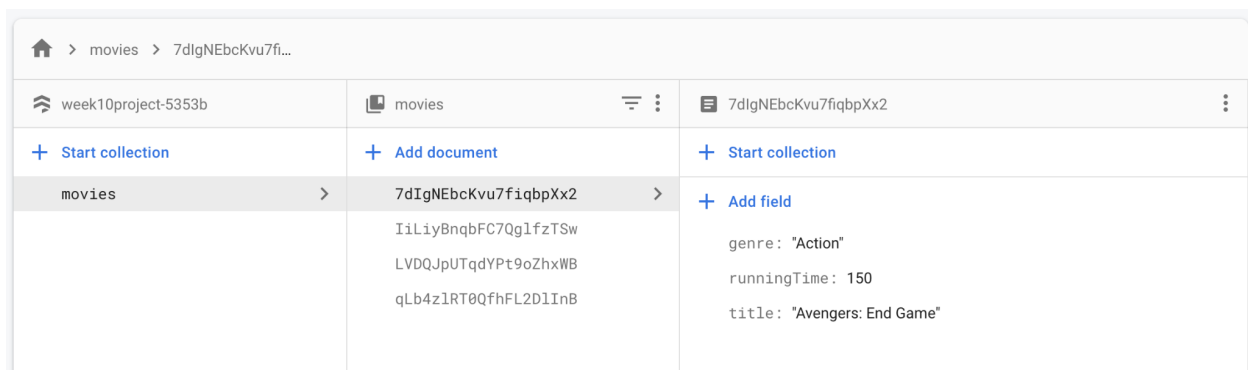
Field	Type	Value	
title	= string	Avengers: End Game	—
genre	= string	Action	—
runningTime	= number	150	—
<div>+</div>			

Cancel

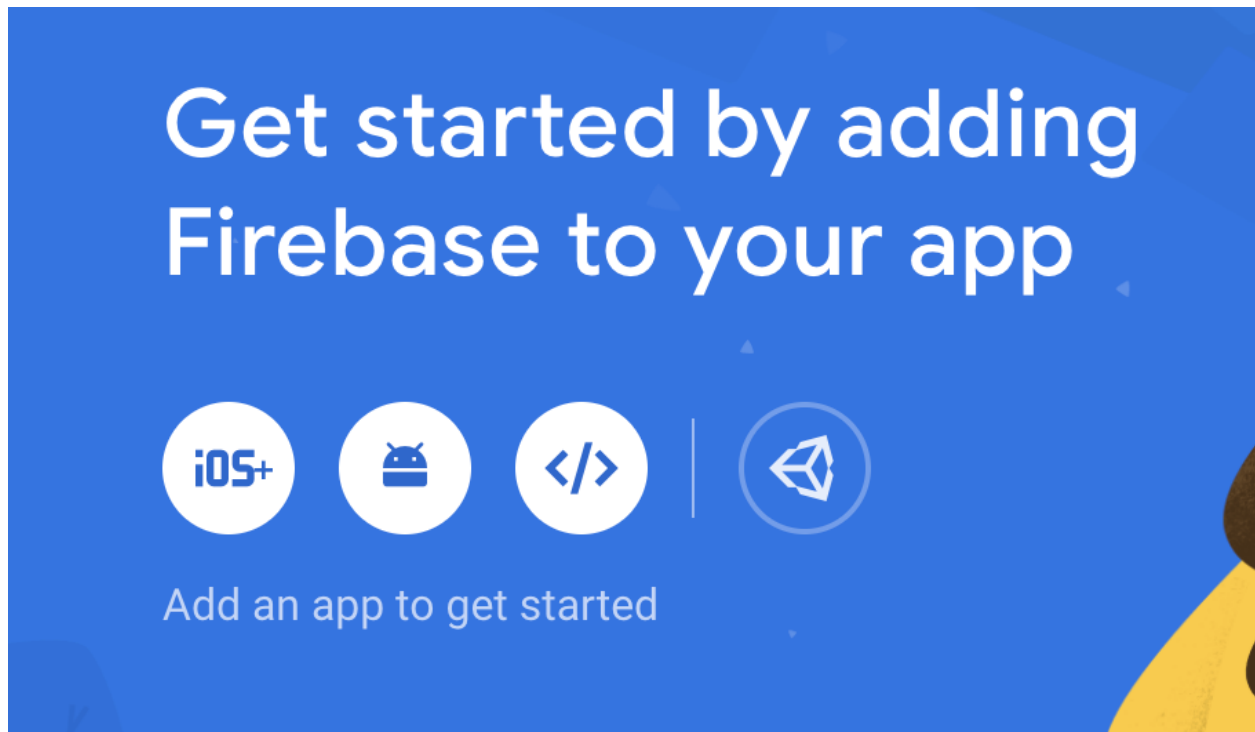
Save

Document gets added to the interface

- You can also add additional documents by tapping the **Add Document** button

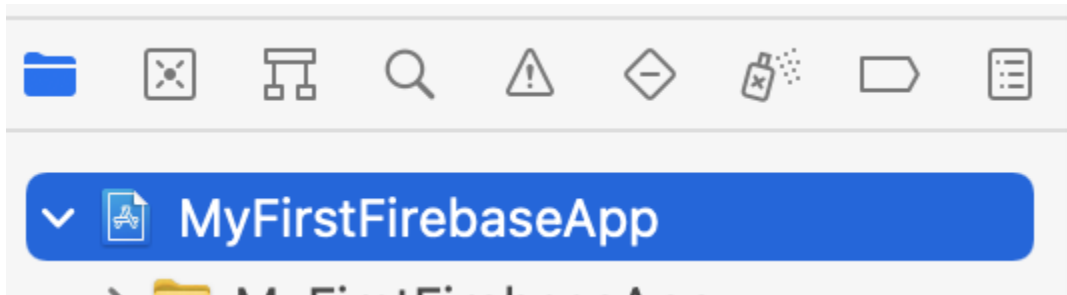


Add



In your IOS application, get the bundle identifier

1/ Click on the project



2/ Get the bundle identifier:

In the example below, the bundle identifier is *com.test.MyFirstFirebaseApp*



Paste your iOS bundle ID into the Firebase wizard:

×

Add Firebase to your Apple app

1

Register app

Apple bundle ID ?

com.test.MyFirstFirebaseApp

App nickname (optional) ?

My Apple app

App Store ID (optional) ?

123456789

Register app

Download the GoogleService-Info.plist file, and add it to your project

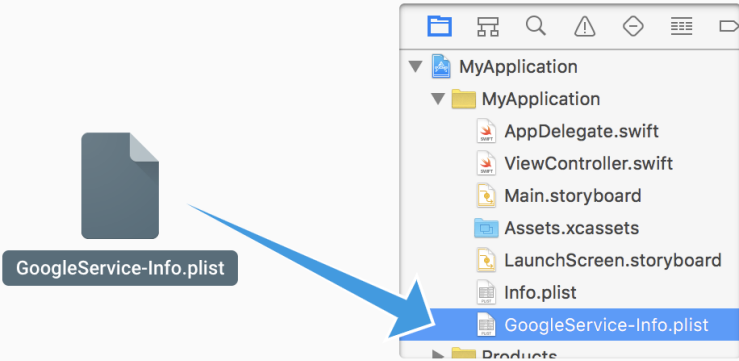
- Ensure that you delete any preexisting GoogleService-Info.plist files from the project!

✓ Register app
Apple bundle ID: com.test.MyFirstFirebaseApp

2 Download config file
Instructions for Xcode below | [Unity](#) [C++](#)

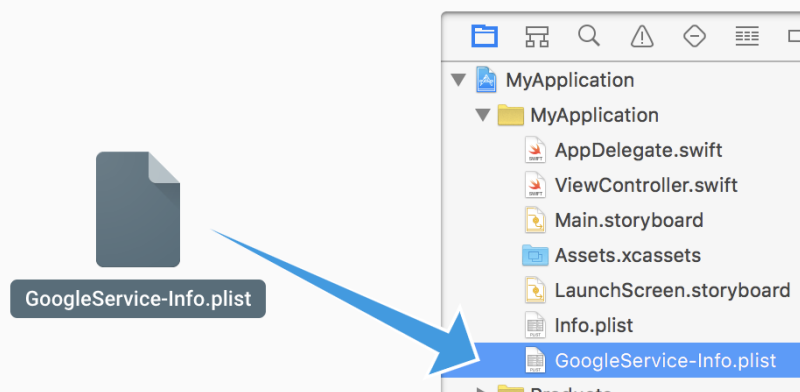
Download GoogleService-Info.plist

Move the GoogleService-Info.plist file that you just downloaded into the root of your Xcode project and add it to all targets.



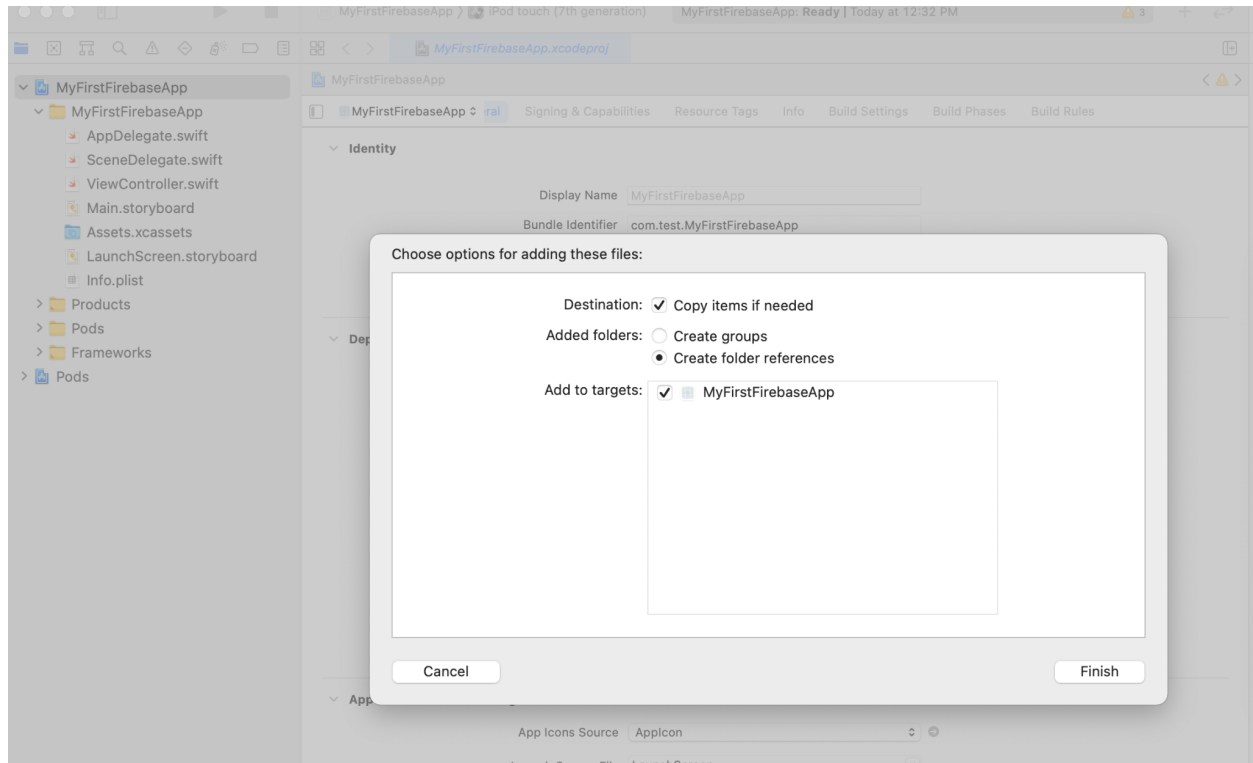
Next

Move the GoogleService-Info.plist file that you just downloaded into the root of your Xcode project and add it to all targets.

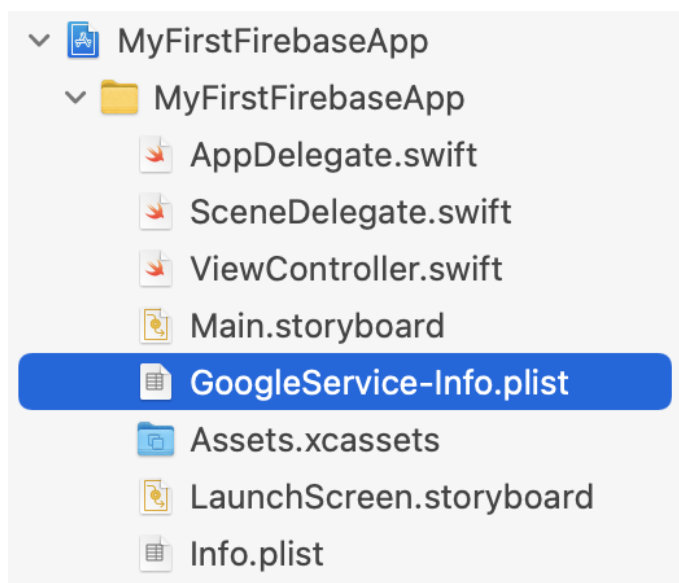


Delete any pre-existing GoogleService-Info.plist files, and then add the one you downloaded from your own Firestore.com account.

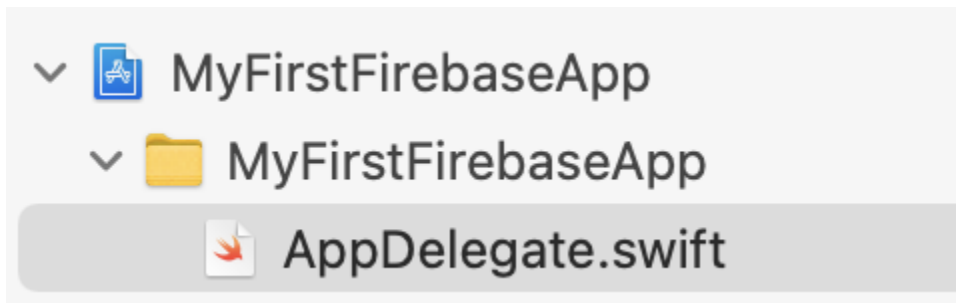
When you drag and drop into the project, make sure you select **copy items if needed**



Result:



In the AppDelegate.swift file, add the Firebase configuration code:



```
import UIKit
import Firebase // TODO: you need to add this!

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.
        FirebaseApp.configure() // TODO: you need to add this
        return true
    }

    // Other boilerplate code is here
}
```

Return to the Firestore console and finish the wizard;

5

Next steps

You're ready!

Make sure that you take a look at the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

Previous

Continue to the console

After the screen refreshes, you will be returned to the main project page.
Your IOS application will appear on this screen:

Week10Project

Spark plan

ios+ com.test.MyFirstFi...

+ Add app

Now, you're ready to code!

Retrieving Data From Firestore

Prerequisites:

- Firebase libraries are added to the project
- Project contains your GoogleInfo.plist file
- AppDelegate file contains the import Firebase and Firebase.configure() statements

1/ Add UI to let user add and retrieve data from Firestore.

```
import UIKit
```

```
class ViewController: UIViewController {
```

```
    // MARK: Outlets
```

```
    @IBOutlet weak var txtRunningTime: UITextField!
```

```
    @IBOutlet weak var txtMovieTitle: UITextField!
```

```
    override func viewDidLoad() {
```

```
        super.viewDidLoad()
```

```
        // Do any additional setup after loading the view.
```

```
    }
```

```
    // MARK: Actions
```

```
    @IBAction func getAllPressed(_ sender: Any) {
```

```
        // Query the firestore collection and return the results
```

```
    }
```

```
    @IBAction func saveDataPressed(_ sender: Any) {
```

```
        // Get the data from the ui
```

```
        // Save this data Firestore
```

```
    }
```

```
}
```

2/ Import Firestore SDK & create a programmatic reference to your firestore database

```
import UIKit
// Add this import to any screen that needs to connect to Firestore
import FirebaseFirestore

class ViewController: UIViewController {

    // MARK: Outlets
    @IBOutlet weak var txtRunningTime: UITextField!
    @IBOutlet weak var txtMovieTitle: UITextField!

    // MARK: Firestore variables
    // This db variable creates a connection to the Firestore database
    // And is used to query & save & access the Firestore database
    let db = Firestore.firestore()

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }

    // MARK: Actions
    @IBAction func getAllPressed(_ sender: Any) {
        // Query the firestore collection and return the results
    }

    @IBAction func saveDataPressed(_ sender: Any) {
        // Get the data from the ui
        // Save this data Firestore
    }
}
```

3/ Code the getAllPressed() function

```

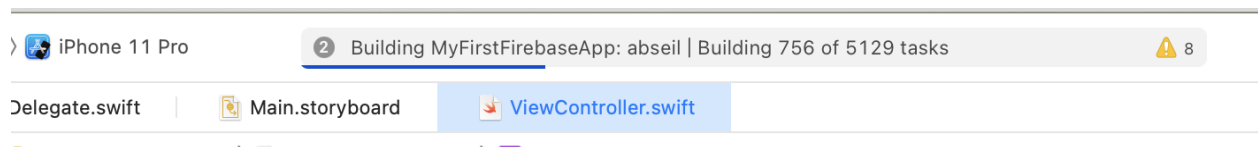
@IBAction func getAllPressed(_ sender: Any) {
    // Query the firestore collection and return the results
    db.collection("movies").getDocuments {
        // If queryResults is NOT nil, then it will contain an array of your documents from
        Firestore
        // if error is NOT nil, then it will contain the error message
        (queryResults, error) in

        // error handling
        if let err = error {
            print("Error getting documents from collection")
            print(err)
            return
        }

        // everything was ok
        if (queryResults!.count == 0) {
            print("No documents found in the movies collection")
        }
        else {
            print("I found results: \(queryResults!.count)")
        }
    }
}

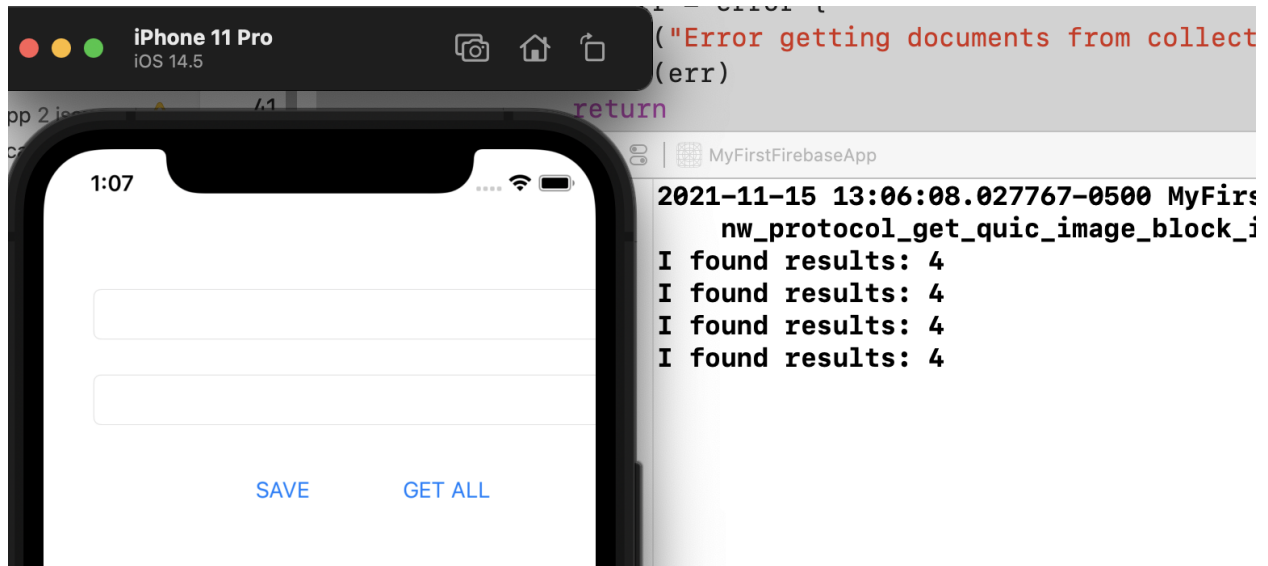
```

The first time you compile the app, it might be really slow:



Expected result:

- When you tap the GET ALL button you see these results
- The number of results should match the number of documents you have in your firestore



My firestore at this point:

<div> movies </div> <div> + Add document </div> <div> <div>7dIgNEbcKvu7fiqbpXx2</div> <div> <div>IiLiyBnqbFC7Qg1fzTSw</div> <div>LVDQJpUTqdYPt9oZhxB</div> <div>qLb4z1RT0QfhFL2D1InB</div> </div> </div>	<div> <div>7dIgNEbcKvu7fiqbpXx2</div> <div> <div>+ Start collection</div> <div>+ Add field</div> <div>genre: "Action"</div> <div>runningTime: 150</div> <div>title: "Avengers: End Game"</div> </div> </div>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Get the details of each document

```

@IBAction func getAllPressed(_ sender: Any) {
    // Query the firestore collection and return the results
    db.collection("movies").getDocuments {
        // If queryResults is NOT nil, then it will contain an array of your documents from
        Firestore
        // if error is NOT nil, then it will contain the error message
        (queryResults, error) in

        // error handling
        if let err = error {
            print("Error getting documents from collection")
            print(err)
            return
        }

        // everything was ok
        if (queryResults!.count == 0) {
            print("No documents found in the movies collection")
        }
        else {
            print("I found results: \(queryResults!.count)")

            // actually look at the results
            // queryResults = array of documents
            for document in queryResults!.documents {
                print("Id:\(document.documentID)")
                print("Contents:\(document.data())")
            }
        }
    }
}

```

Expected result:

- You should be able to see the individual contents of each document in the Firestore collection

```
2021-11-15 13:14:39.232850-0500 MyFirstFirebaseApp[56812:3486752] []
    nw_protocol_get_quic_image_block_invoke dlopen libquic failed
I found results: 4
Id:7dIgNEbcKvu7fiqbpx2
Contents:["title": Avengers: End Game, "runningTime": 150, "genre": Action]
Id:IiLiyBnqbFC7QglfzTSw
Contents:["runningTime": 45, "title": Squid Game, "genre": Thriller]
Id:LVDQJpUTqdYPt9oZhxB
Contents:["runningTime": 90, "title": Frozen, "genre": Family]
Id:qLb4z1RT0QfhFL2DlInB
Contents:["runningTime": 138, "title": Dr. Strange and the Multiverse, "genre":
    Action]
```

Get the individual values from each document

```

@IBAction func getAllPressed(_ sender: Any) {
    // Query the firestore collection and return the results
    db.collection("movies").getDocuments {
        // If queryResults is NOT nil, then it will contain an array of your documents from
        Firestore
        // if error is NOT nil, then it will contain the error message
        (queryResults, error) in

        // error handling
        if let err = error {
            print("Error getting documents from collection")
            print(err)
            return
        }

        // everything was ok
        if (queryResults!.count == 0) {
            print("No documents found in the movies collection")
        }
        else {
            print("I found results: \(queryResults!.count)")

            // actually look at the results
            // queryResults = array of documents
            for document in queryResults!.documents {
                print("Id:\(document.documentID)")
                print("Contents:\(document.data())")

                let item = document.data()
                // using nil coelasing to deal with the fact that some keys may be null
                print("Title: \(item["title"] ?? "N/A")")
                print("Running Time: \(item["runningTime"] ?? "N/A")")
                print("Genre: \(item["genre"] ?? "N/A")")
                print("-----")
            }
        }
    }
}

```

Expected result

```

Id:7dIgNEbcKvu7fiqbpXx2
Contents:["genre": Action, "title": Avengers: End Game, "runningTime": 150]
Title: Avengers: End Game
Running Time: 150
Genre: Action
-----
Id:IiIiyBnqbFC7QglfzTSw
Contents:["title": Squid Game, "genre": Thriller, "runningTime": 45]
Title: Squid Game
Running Time: 45
Genre: Thriller
-----
Id:LVDQJpUTqdYPt9oZhxB
Contents:["title": Frozen, "genre": Family, "runningTime": 90]
Title: Frozen
Running Time: 90
Genre: Family
-----
Id:qLb4z1RT0QfhFL2DlInB
Contents:["title": Dr. Strange and the Multiverse, "genre": Action, "runningTime":
138]
Title: Dr. Strange and the Multiverse
Running Time: 138
Genre: Action
-----

```