AOZ Firebase Quick Start Guide

By Phil Bell April 2022

https://aozwithphil.com/

phil@aozwithphil.com

Contents

What is Firebase?

Firebase is Google's mobile development platform which is backed by the Google Cloud Platform.

The Firebase platform is a backend-as-a-service solution both for mobile and web-based applications that includes services for building, testing, and managing apps.

The AOZ Firebase instructions provide access to the following Firebase services: -

Cloud Firestore is a flexable, scalable NoSQL cloud database to store and sync data for client and server-side development.

Cloud Storage for Firebase allows the secure storage of files, such as photos and documents.

Authentication provides an end-to-end identity solution, supporting email and password accounts, phone auth, and Google, Twitter, Facebook, and GitHub login, and more.

Read more about Firebase at https://firebase.google.com/

Is Firebase Free?

Firebase has 2 billing plans, Spark and Blaze.

The Spark plan is free and can be used at no cost with very generous usage limits to get you started.

The Blaze plan is Pay-As-You-Go and includes free usage limits provided with the Spark plan.

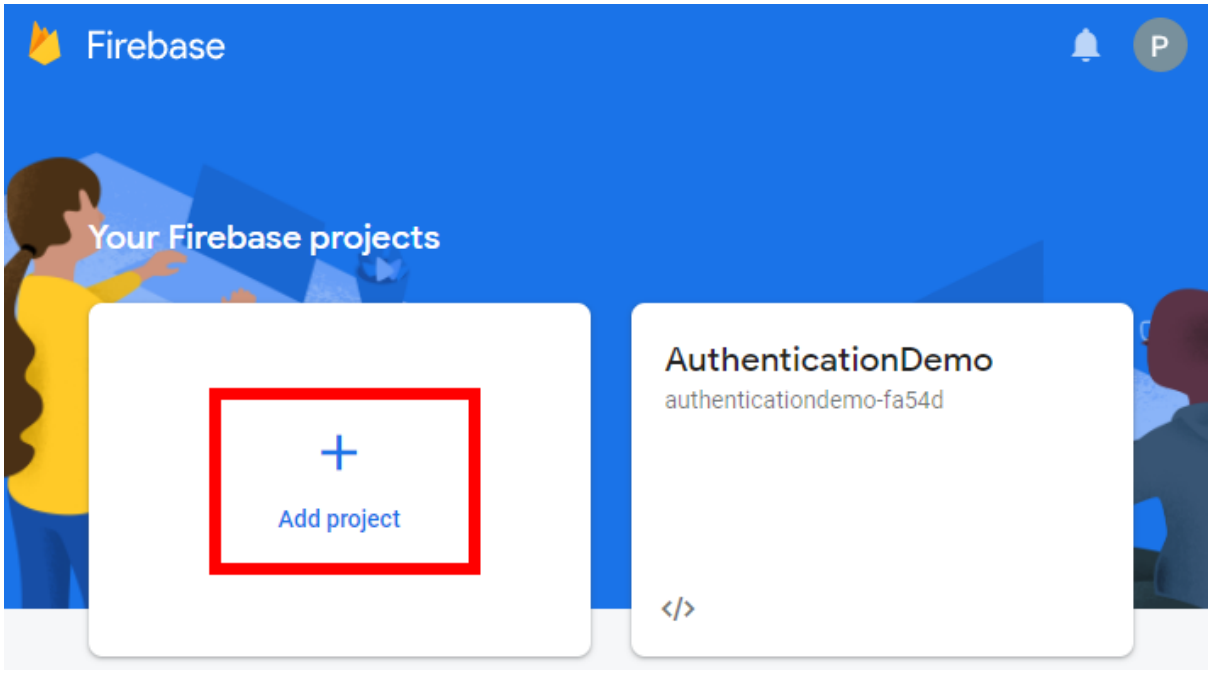
Further details at https://firebase.google.com/pricing

Create your first Firebase project

Before you can begin using the AOZ Firebase instructions you will need to create your project in the Firebase web console.

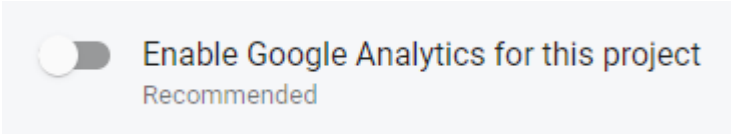
You can login to the console here: -

https://console.firebase.google.com/



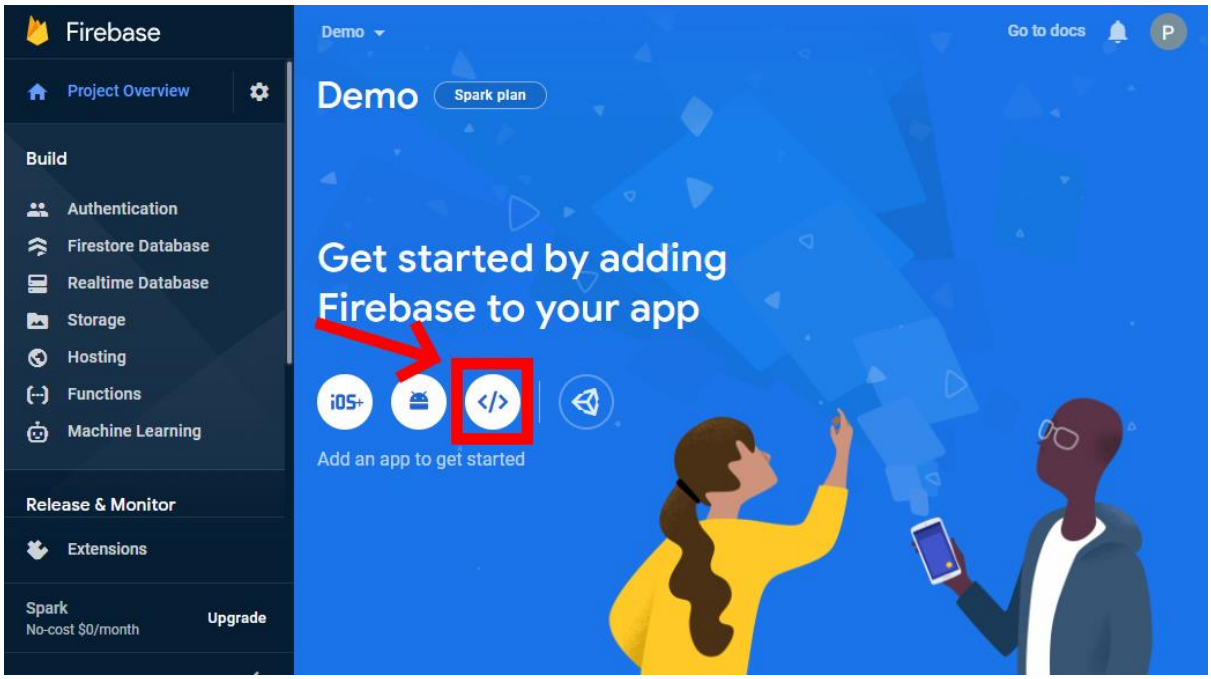
Click the Add project and follow the steps to create your project.

In step 2 there is an option to enable Google Analytics, this is optional and can be disabled.



Once the project has been created you will be taken to the Project Overview page where you can create your web app for your AOZ Application.

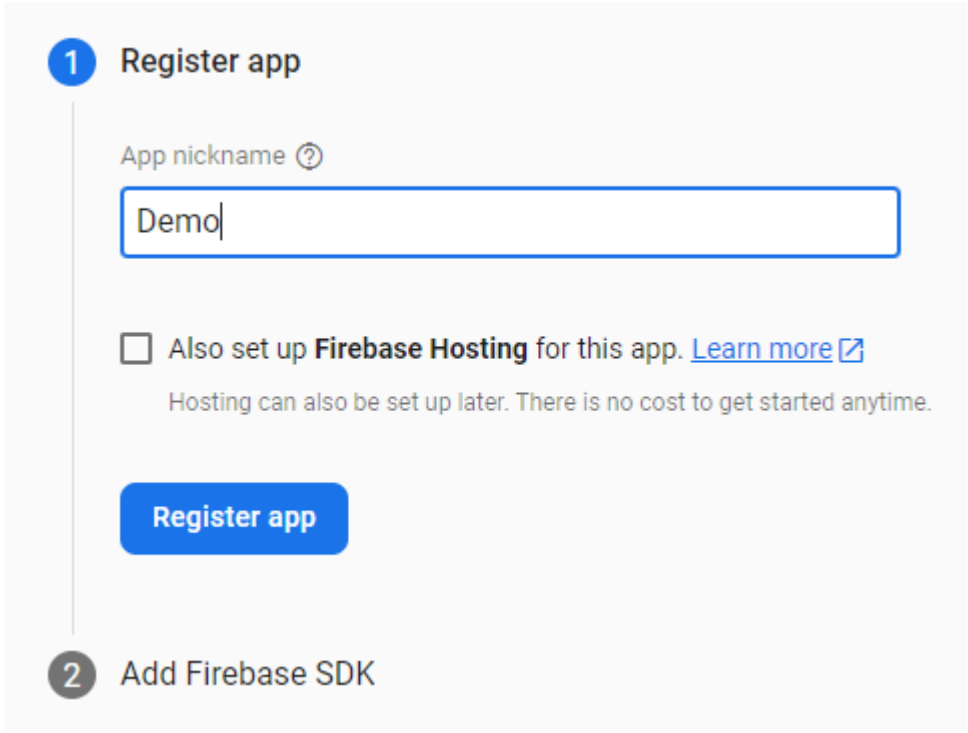
Create your web app



Click the web icon highlighted above

Enter the App nickname for your application then click the Register app button

You can leave the Firebase Hosting option unselected at this stage.



Click the Register app button to continue.

The Firebase API configuration will then be provided: -

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyDOLnmec_xLcxMdZRLudaLSTQVBn7KQPtM",
  authDomain: "fir-aa540.firebaseapp.com",
  projectId: "fir-aa540",
  storageBucket: "fir-aa540.appspot.com",
  messagingSenderId: "229056896501",
  appId: "1:229056896501:web:73b418f87f9ac2ef0f94bc"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

Note: This option uses the modular JavaScript SDK, which provides reduced SDK size.

Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples

Continue to console

Select and copy the highlighted items shown above, these values are required to initialise Firebase in your AOZ application with the **Firebase Init** instruction.

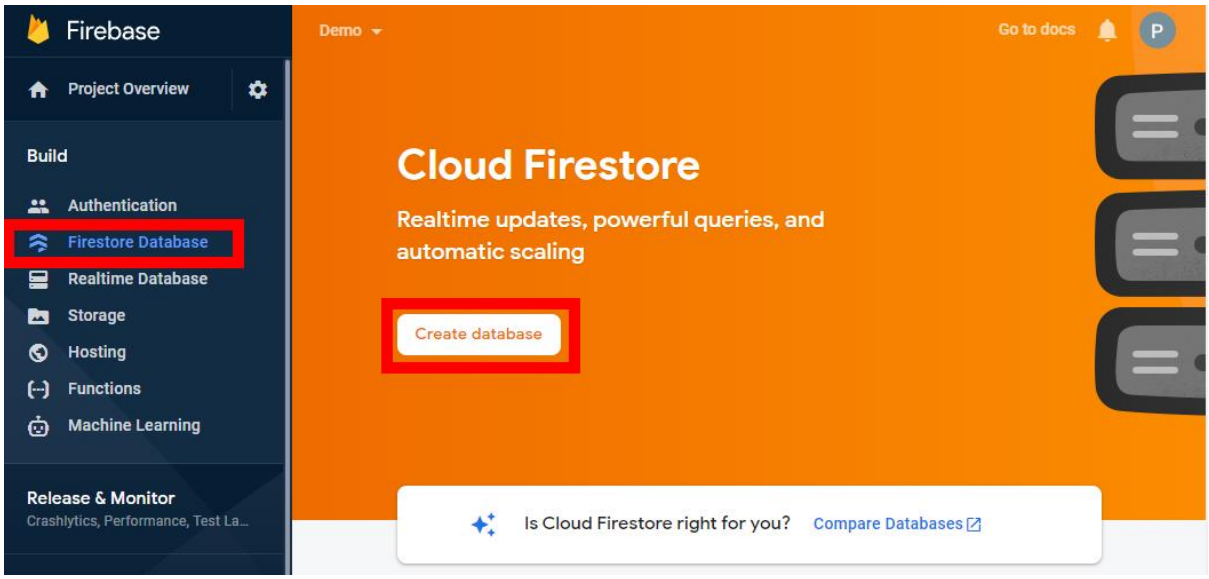
You can then click the Continue to console button.

Add the **Firebase Init** instruction to the top of your AOZ Project code window and fill in the parameters you copied above.

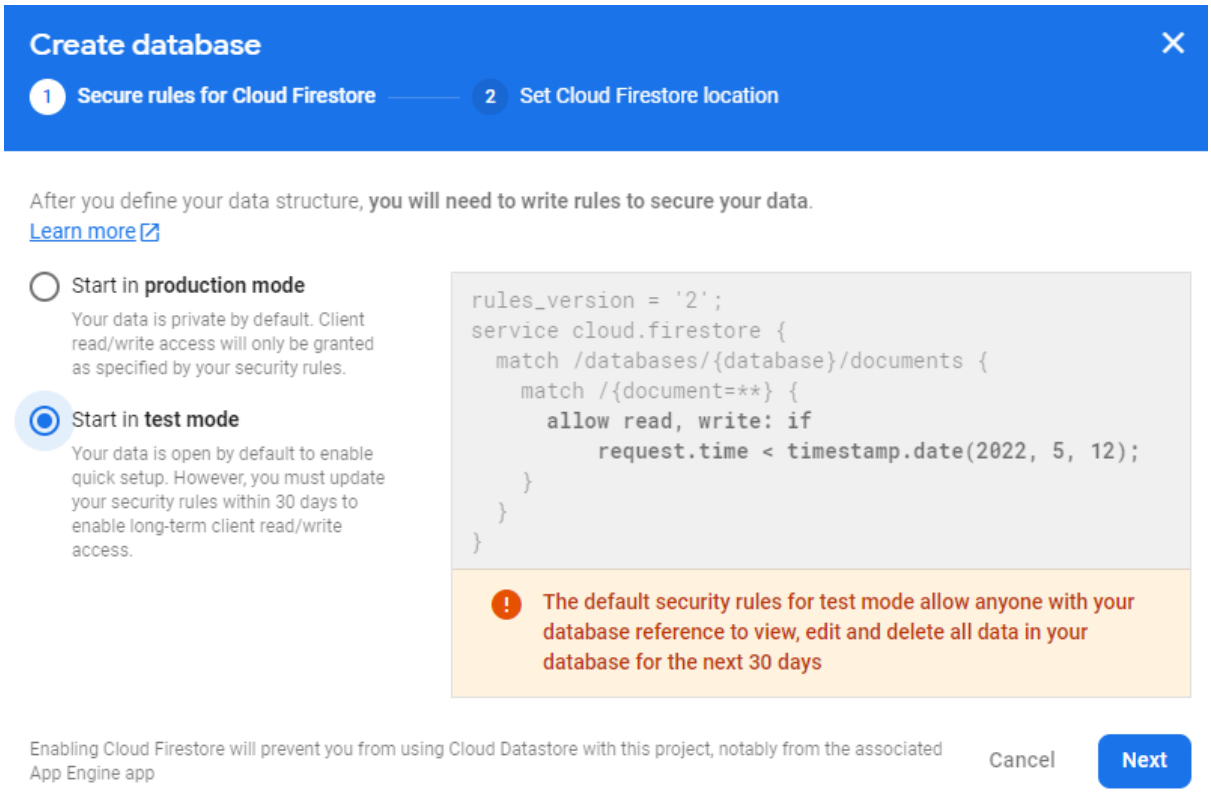
Firebase Init apiKey$="", authDomain$="", projectId$="", storageBucket$="", messagingSenderId$="", appId$=""

Configure Firestore Database

Select the Firestore Database from the left navigation then click the Create database button.

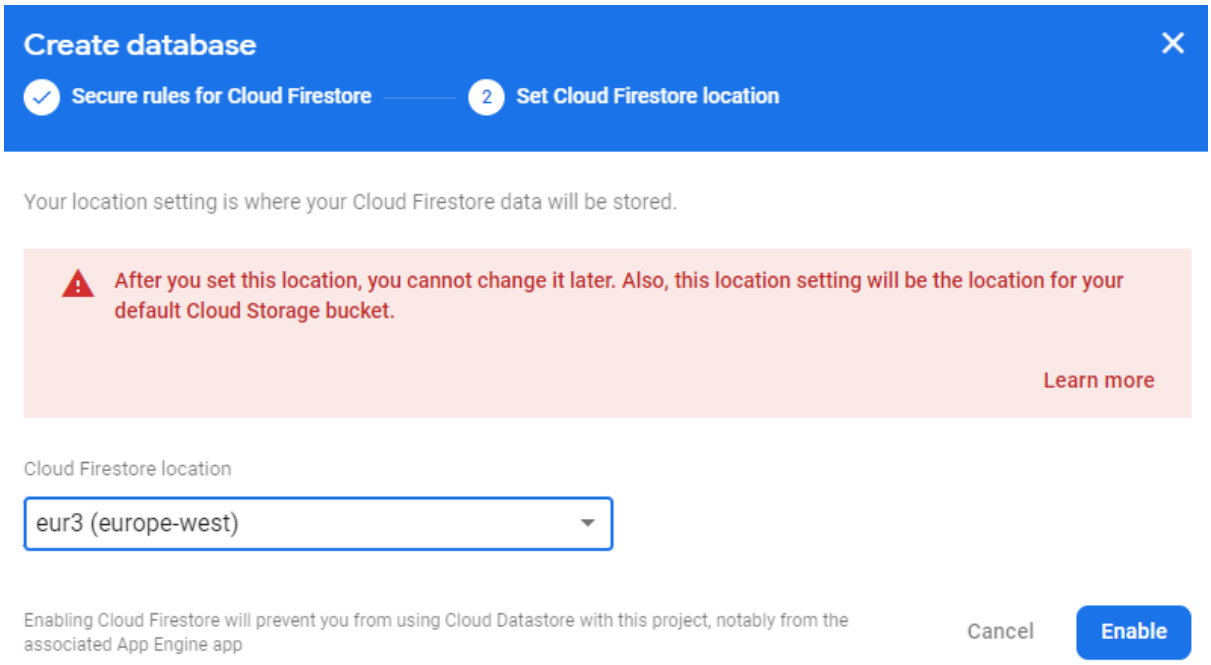


Select the **Start in test mode** option, as shown below this will allow unauthenticated access to your database for 30 days. You can at any point update the security rules as and when required.



Read more about Firestore security rules here
https://firebase.google.com/docs/firestore/security/get-started

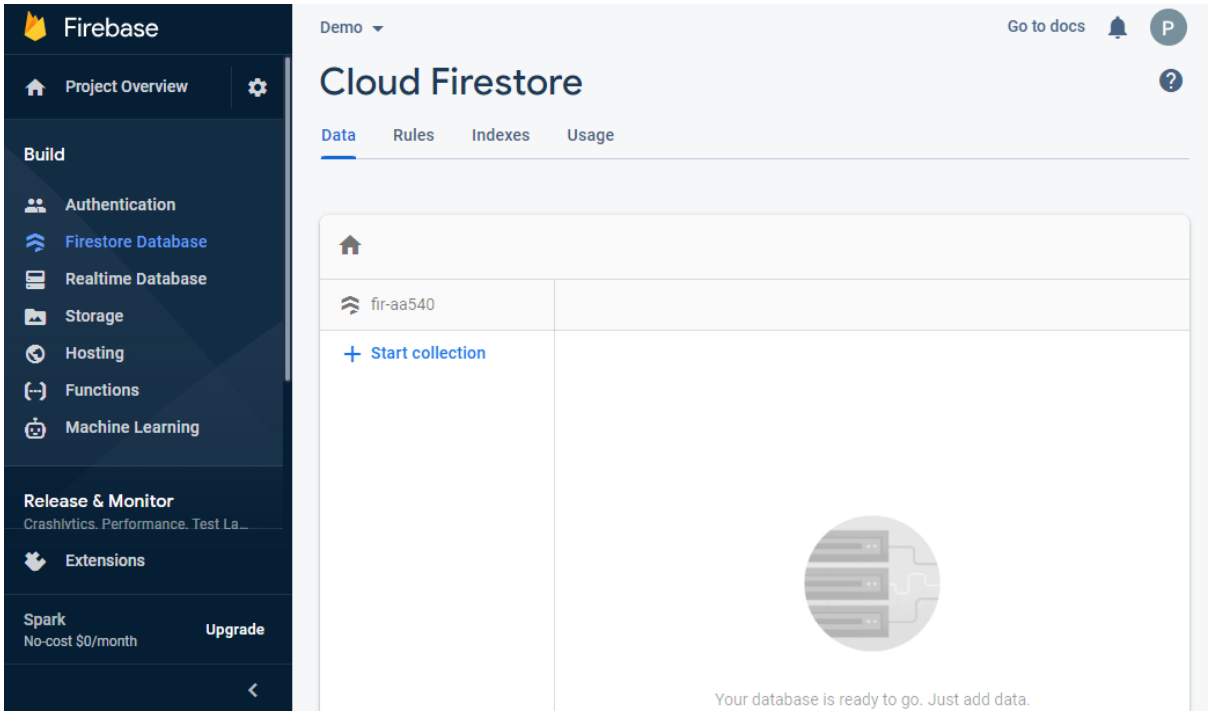
Next, select the region you wish your database to be located, then click the Enable button



You will then be shown the Firebase Database data explorer interface.

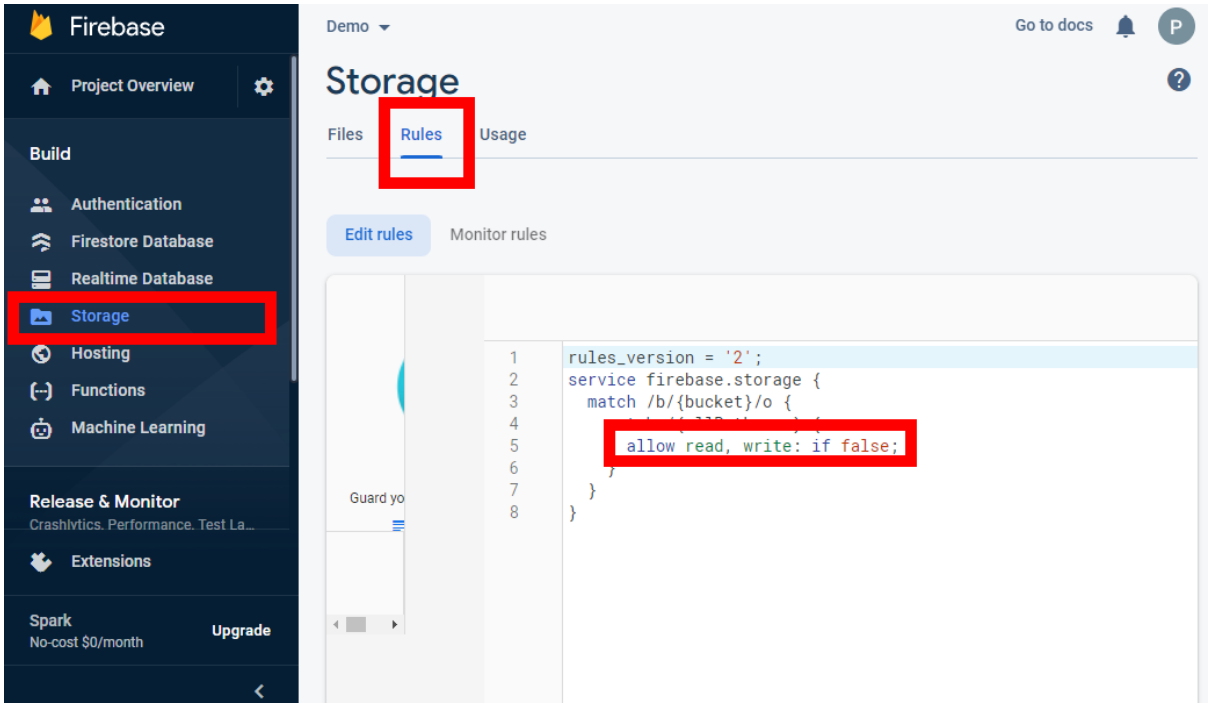
Here you can create and manage your data if required, this is not necessary as the AOZ Firebase instructions allow you to create and access your collection data.

This web interface is very useful when developing your application as you can easily check your data is being written as expected. You can also add, delete and update your application data directly.



Configure Firebase Storage

Uploading files to Firebase storage is by default disabled with a security rule. Before you can upload files using the AOZ Firebase instructions, you will need to update the security rules.



To allow read/write access you can change the rule highlighted above to: -

allow read, write: if true;

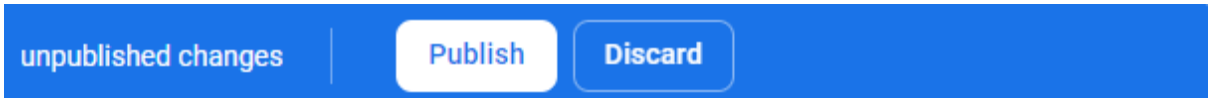
However, this will allow unauthenticated access to your file storage, which is ok during development, but it is highly recommended that you secure your storage by only allowing authenticated users.

This can be achieved using the following rule: -

allow read, write: if request.auth != null;

This rule can also be used to secure your Firestore database.

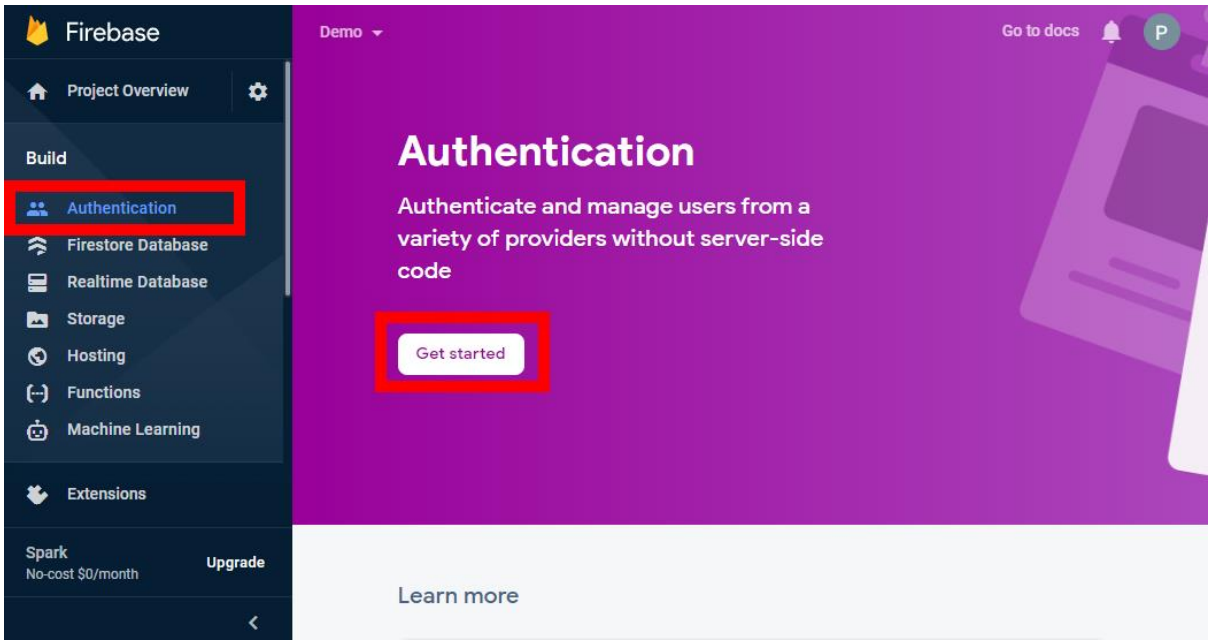
To save your changes click the Publish button.



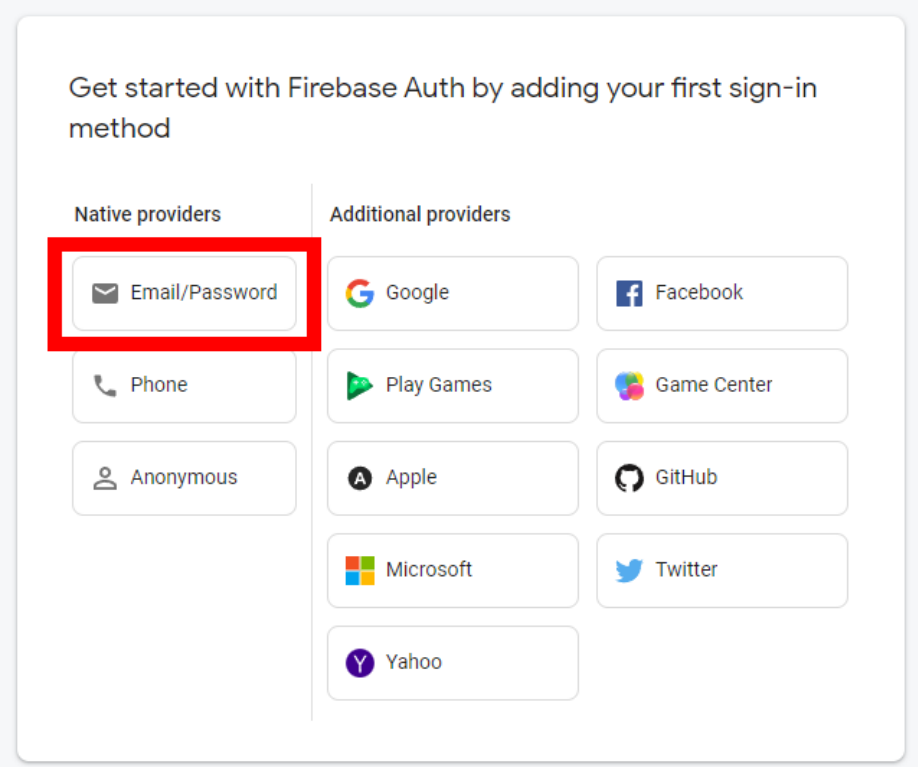
Configure Firebase Authentication

To use the AOZ Firebase Authentication instructions you must first configure the sign-in-methods you wish to use for your application.

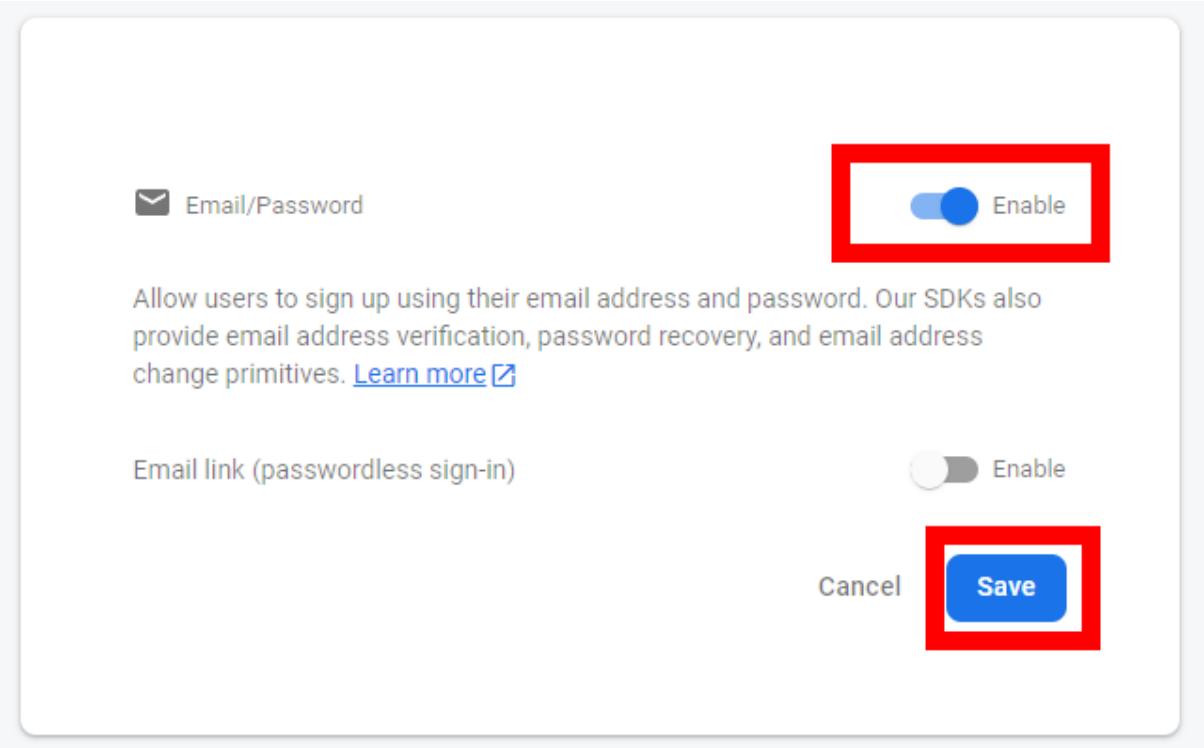
Select the Authentication item from the left navigation panel, then click the Get started button.



The most common use case would be to provide the Email & Password sign-in method, you can enable this by clicking the button shown below:-



Enable the provider then click the save button

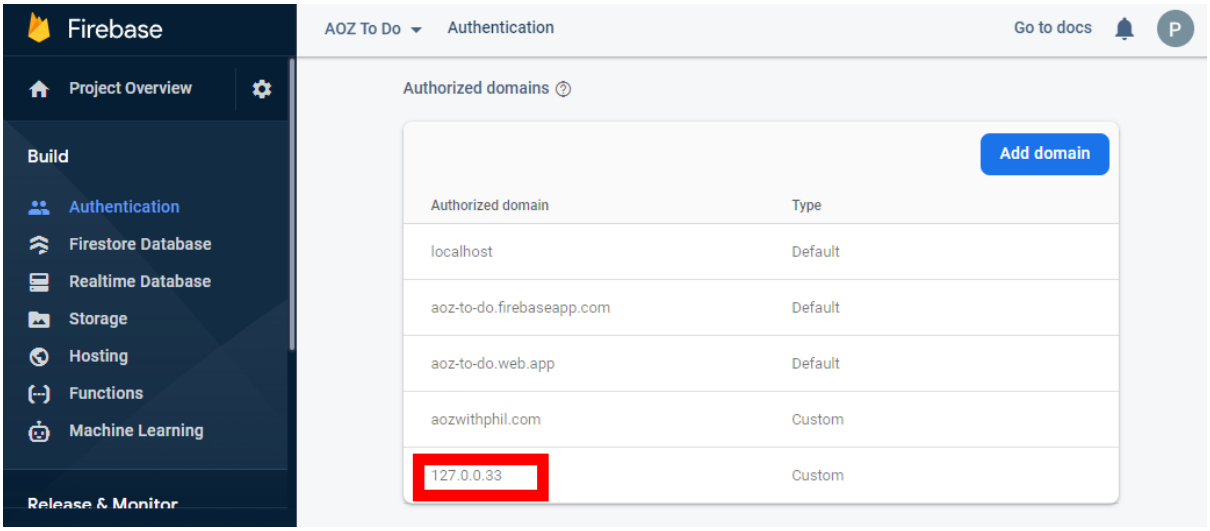


Additional configuration is required to use the other 3rd party providers such as Google, Facebook, GitHub etc

For example, the GitHub provider requires you to provide a Client ID and Client Secret. These can be created via the providers own web site. You will need to consult their documentation for instructions on how to create and manage these credentials.

When using a 3rd party sign in method you will need to authorise the domain for OAath redirects.

For local development you should add **127.0.0.33** as shown below: -



AOZ Authentication Example

Here is a basic example of how to implement authentication in your AOZ application.

It uses a built-in user interface and uses both email/password and Google sign-in providers

```
// Fill in the parameters for your application
Firebase Init apiKey$="", authDomain$="", projectId$="", storageBucket$="", messagingSenderId$="", appId$=""


If FirebaseAuth AuthState() = false
        // User is not signed in
        Goto SIGN_IN
End If

MAIN:
        Print "User Signed In"
        Print FirebaseAuth UserEmail$()
        Print FirebaseAuth UserDisplayName$()
        Do
                If FirebaseAuth AuthState() = false
                        // User is not signed in or has signed out
                        Goto SIGN_IN
                End If

                Wait Vbl
        Loop

SIGN_IN:
        FirebaseAuth SignInWithBuiltInUI, provider$="google,email"

        // Wait here until the user is signed in
        Do
                If FirebaseAuth AuthState() = true
                        Goto MAIN
                End If

                Wait Vbl
        Loop
```

Understanding the Cloud Firestore Database

The Cloud Firestore database is a flexible, scalable NoSQL cloud database to store and sync data for client- and server-side development.

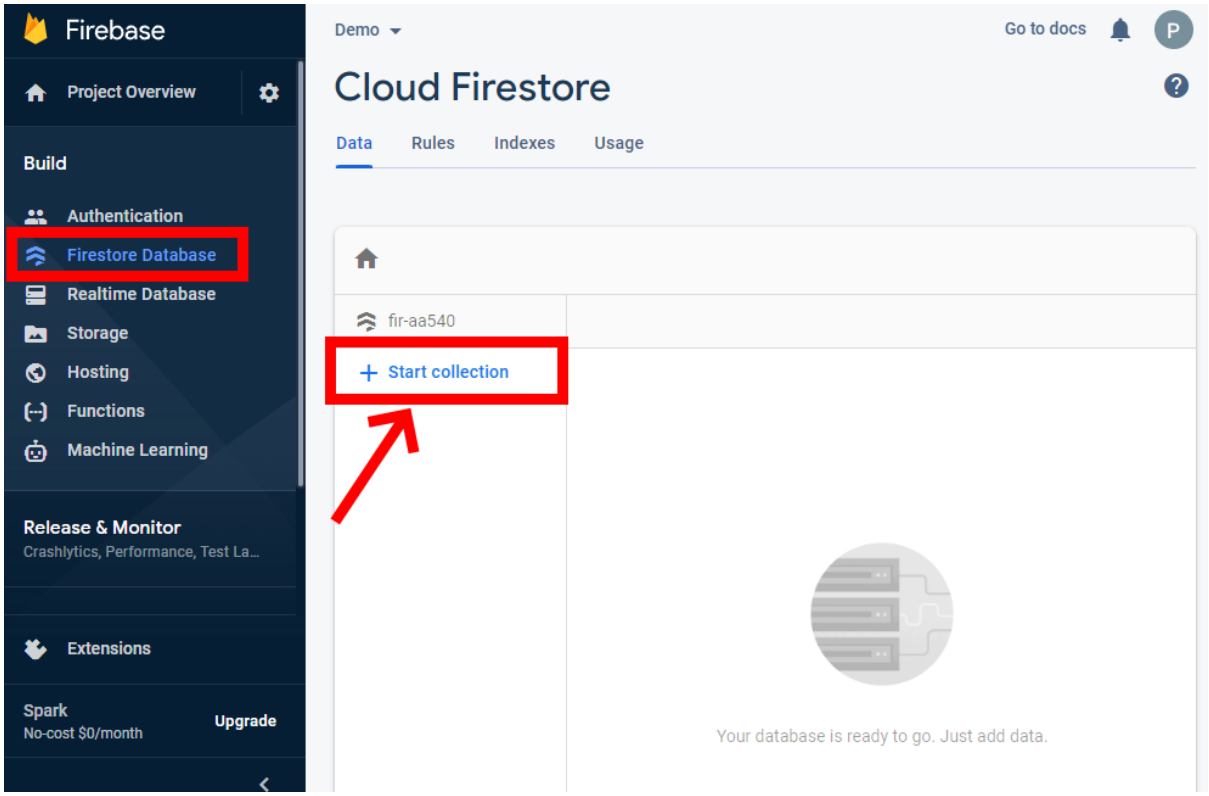
Following Cloud Firestore's NoSQL data model, you store data in documents that contain fields mapping to values. These documents are stored in collections, which are containers for your documents that you can use to organize your data and build queries.

Documents support many different data types, from simple strings and numbers, to complex, nested objects. Unlike a SQL database, there are no tables or rows.

Further information: https://firebase.google.com/docs/firestore/data-model

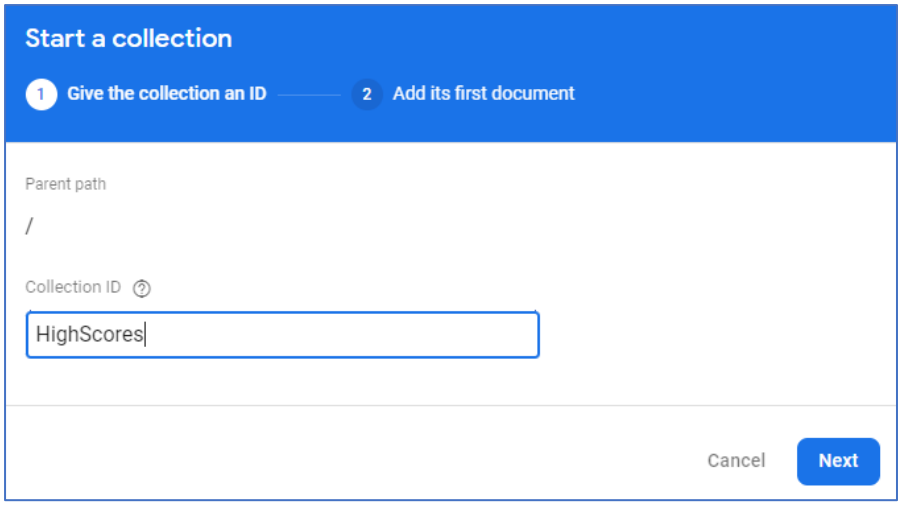
Using the Firebase console to create and manage data

https://console.firebase.google.com/

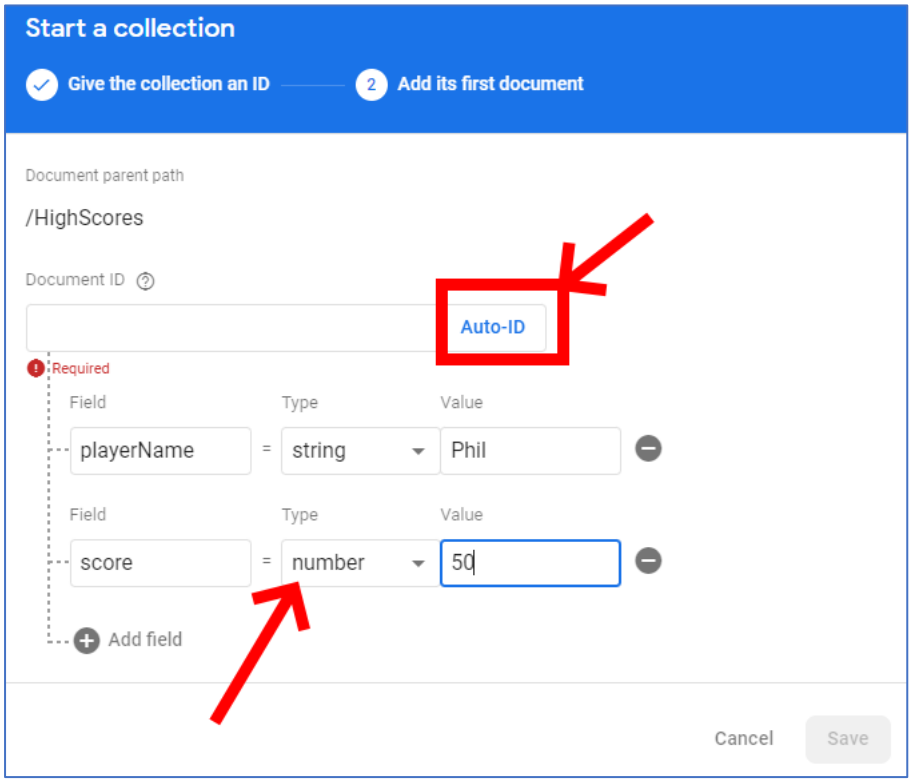


When you create your Firebase project your Firestore database will be empty. You can use the AOZ Firebase instructions to begin creating your data, but you can also use the Firebase console to view and manage your data.

Let's create a **HighScore** collection, click the "Start Collection" button.



Enter "**HighScores**" for the collection name and click next.

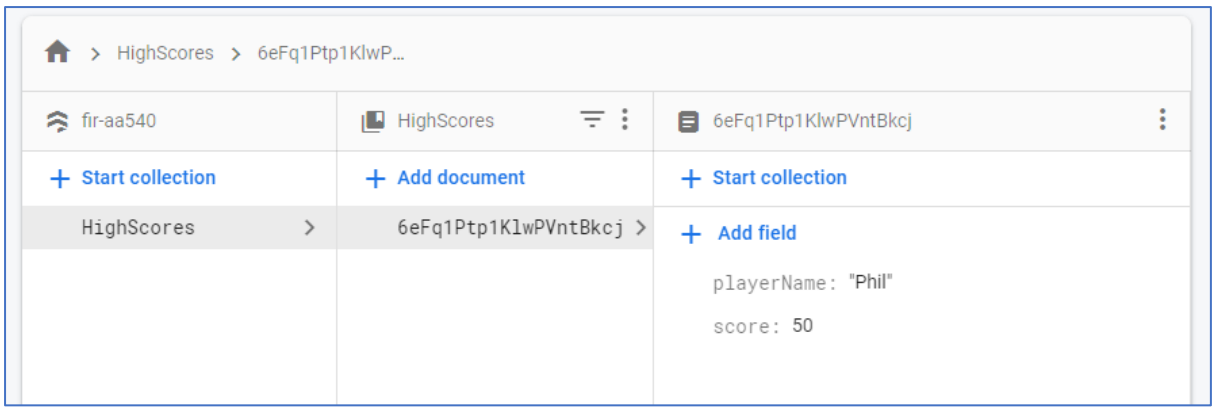


Each document in a Firestore collection must have a unique Document ID. You will use the Document ID extensively in your AOZ applications when using the Firebase instructions to read and update your collection documents.

You can enter a Document ID yourself, but it is best to let Firestore create one by clicking the "**Auto-ID**" button. The auto generated Document ID is a unique alpha numerical string and is 20 characters in length.

When adding fields, you should ensure you select the correct field **Type** for the data you are going to be storing. In the example above, the score field will be numerical, so we have selected **number** as the field **type**.

Add the fields shown above, then click the Save button.



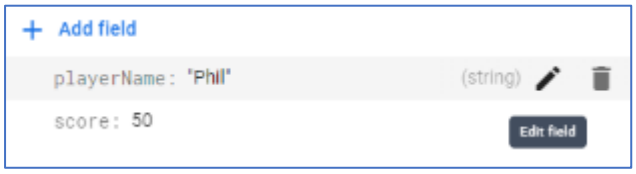
The **HighScores** collection now has one document with a Document ID of **6eFq1Ptp1KlwPVntBkcj**

You can add more documents to the **HighScores** collection by clicking the **Add document** button.

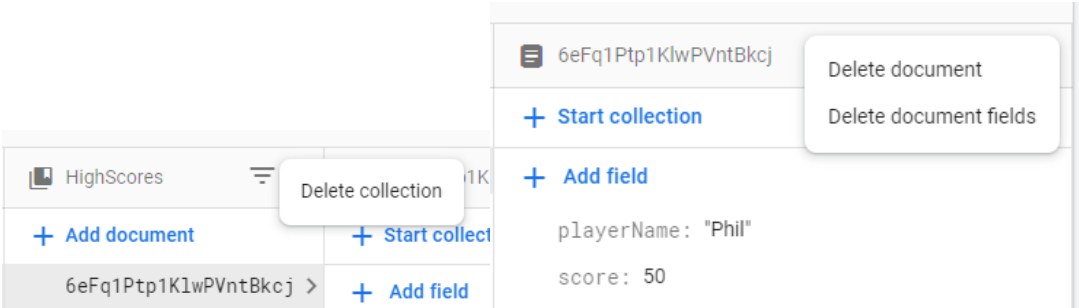
You will notice that you will need to add the **playerName** and **score** fields as we did previously. This is because the documents in a collection do not need to have the exact same fields defined for each document.

Unlike an SQL database where the table structure must be defined upfront and before adding data, each document in the Firestore database collection can have completely different fields to the other documents in the same collection.

The console allows you to add and delete fields in an existing document, and you can edit the field values. Hover your mouse over the field to reveal the edit and delete buttons as shown below: -



You can also delete documents and collections. Click the “3 dots” icon to reveal the context menu: -



During the development of your AOZ application you should use this console on a regular basis to check the data is being stored as you expect. It is also an easy way to create and delete any test data you are using during development.

Add document AOZ code example

Here is an example code snippet to add a document to the **HighScores** collection.

First, we create a new document using the Firebase NewDocument function. An empty document is created and held in memory.

```
docRef$ = Firebase NewDocument$()
```

We then use the `docRef$` as a parameter to set the document fields and their values.

```
Firebase SetDocumentString docRef$ = docRef$, field$ = "playerName", value$ = "Demo Player"
Firebase SetDocumentValue docRef$ = docRef$, field$ = "score", value = 20
```

Next, we add the document to the **HighScores** collection. The **HighScores** collection will be created if it does not already exist.

```
Firebase DocumentAdd collection$ = "HighScores", docRef$ = docRef$, onCompleted$ = "ON_COMPLETED"
```

The procedure named in the `onCompleted$` parameter is called once the document has been written to the Firestore database. You can check if an error has occurred and are also provided the auto generated Document ID which you may need elsewhere in your AOZ application.

```
Procedure ON_COMPLETED[ERROR$, ERROR_CODE, DOCUMENT_ID$, DOCUMENT_REF$]
    If ERROR$ <> ""
        // Handle error
    End If
    // Remove the document from memory as it is no longer needed for this example
    Firebase DocumentDispose DOCUMENT_REF$
End Proc
```

Collection Query AOZ Code example

First, we create a new collection query, a reference to the query is returned by the function.

```
queryRef$ = Firebase NewQuery$ (collection$ = "HighScores", onData$="ON_DATA_RECEIVED")
```

We want the results to be returned with the best score first, so we add an order by clause to the query to ensure the documents are returned in descending order.

```
Firebase QueryAddOrderBy queryRef$ = queryRef$, field$ = "score", direction$ = "desc"
```

Finally, we get the data from the Firestore database, in this example we only want the top 10 scores, so we limit the results by setting the optional **limit** parameter.

```
Firebase QueryGet queryRef$, limit = 10
```

Once the documents have been retrieved, the procedure named in the **onData$** parameter is called.

The procedure receives variables for the number of documents returned, an **ERROR$** variable that can be checked to see if an error occurred, and a reference to the query that can be used to access the documents.

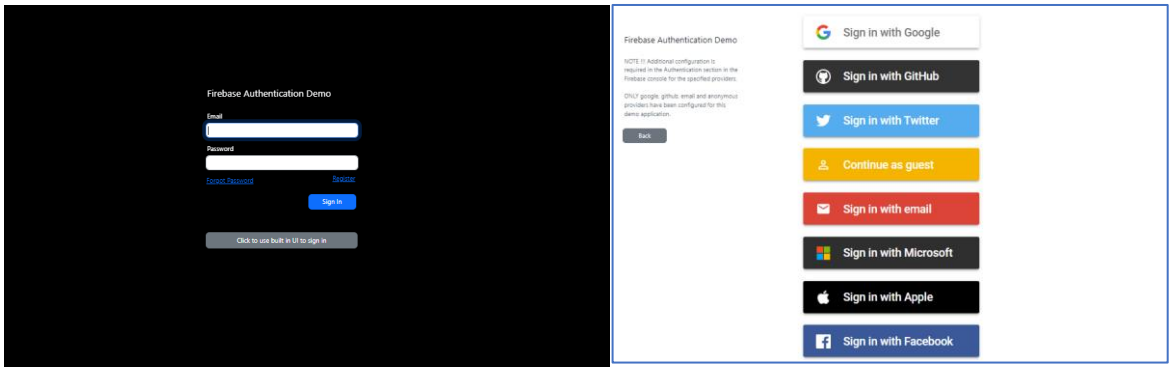
```
Procedure ON_DATA_RECEIVED[DOCUMENT_COUNT, QUERY_REF$, ERROR$, ERROR_CODE]
        For row = 0 To DOCUMENT_COUNT -1
                // Get the Firebase Document ID for a specific row in the query results
                docId$ = Firebase QueryGetDocumentID$(QUERY_REF$, row)
                // Read the document fields using the Firebase Document ID as a reference
                Print Firebase QueryGetString$(QUERY_REF$, docId$, "playerName")
                Print Firebase QueryGetValue(QUERY_REF$, docId$, "score")
        Next row
        // Remove the query and the documents from memory
        Firebase QueryDispose QUERY_REF$
End Proc
```

AOZ Demo Applications

You will find comprehensive demo applications in the AOZ Store and at https://aozwithphil.com/

Firebase Authentication Demo

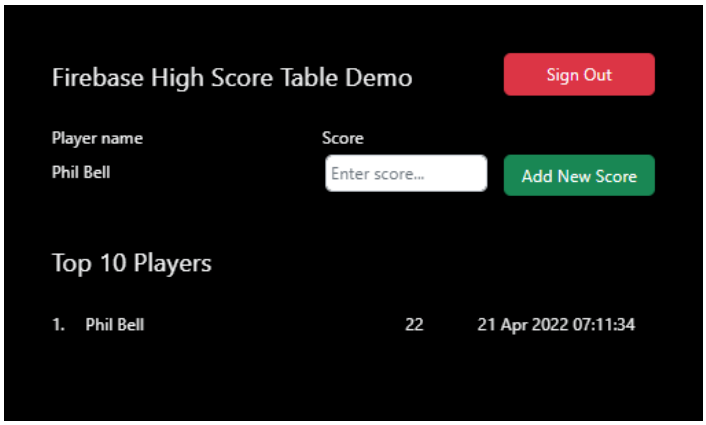
Download: Demo AOZ Project



This demo shows how to build a sign-in/register page for your application using both the built in 3rd party authentication providers and a typical email and password user interface using the AOZ Firebase instructions to create and manage user accounts, sign-in and reset account passwords.

Firebase High Score Table Demo

Download: Demo AOZ Project



A simple demo showing how to create a high score table for your AOZ project. It takes advantage of the real-time updates feature of the Firestore database where changes to the database are automatically sent to the application.

Firebase AOZ Task App Demo

Download: [Demo AOZ Project]

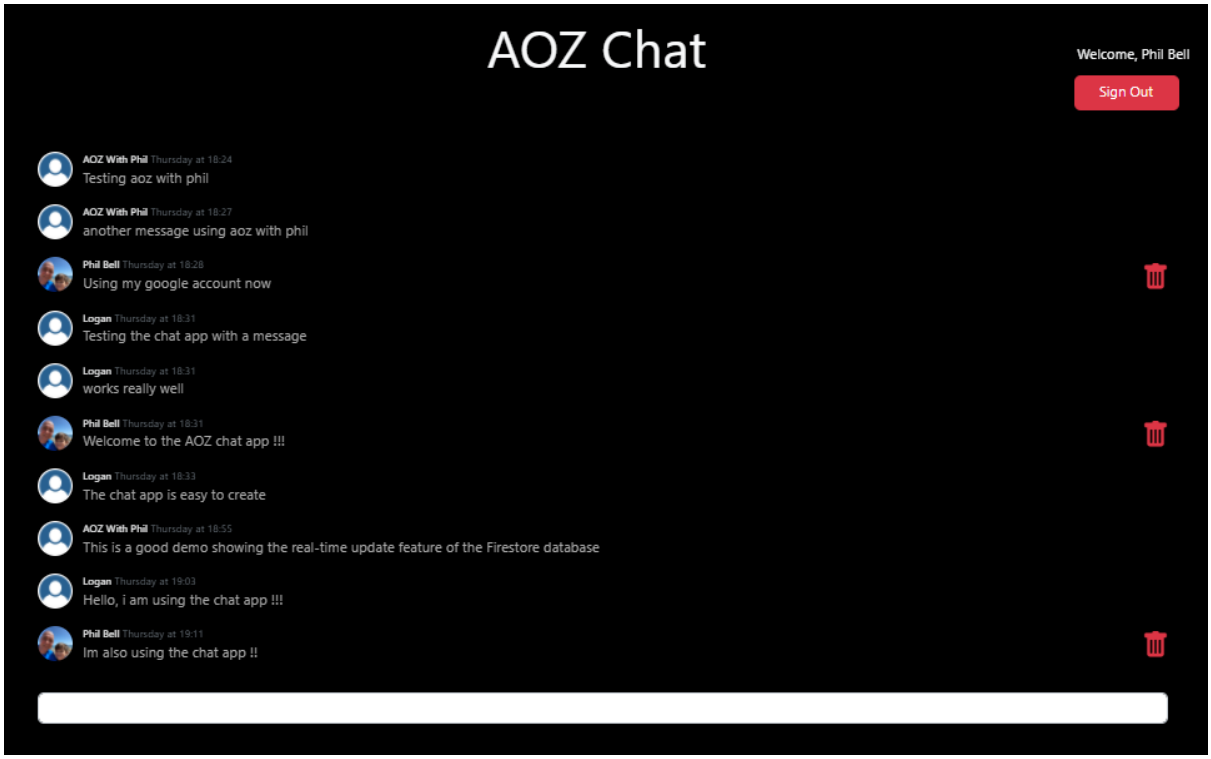
This demo shows how to create a personal task management application.

Reviewing the code, you will learn how to read, create, update, and delete data in your Firestore database using the AOZ Firebase instructions.

The demo also shows how to upload/delete files to Firebase storage.

Firebase AOZ Chat App

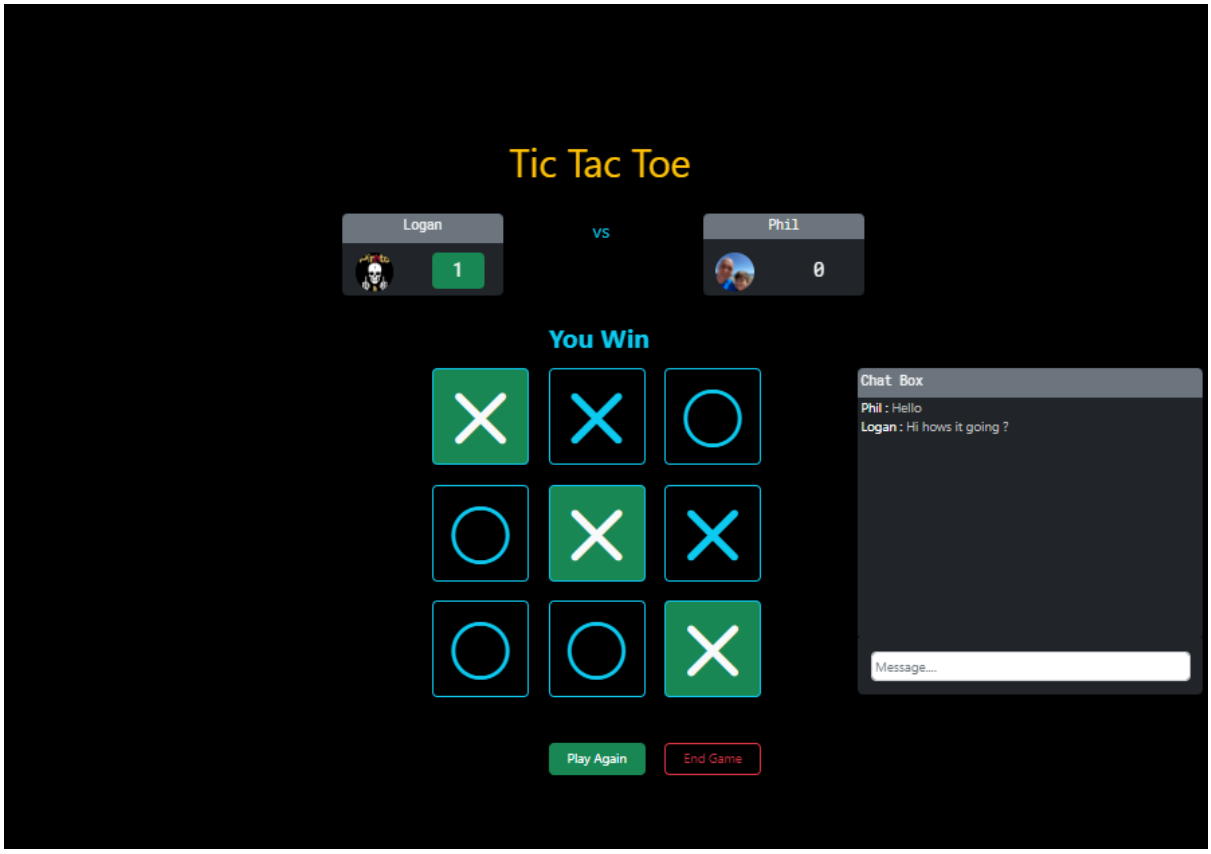
Download: [Demo AOZ Project]



A demo showing how to use the real-time updates feature of the Firestore database to create a simple chat box for your application.

Tic Tac Toe – Multiplayer Turn-Based Game Demo

Download: [Demo AOZ Project]



Uses Google Firebase for player authentication and game state management.