🔥 Firebase Quick Start Guide

Learn how to:

- Set up Firebase project
- Use Firestore database
- Access from JavaScript or Flutter

1.Sign in with a Google Account

1. Visit the Firebase website (firebase.google.com) and sign in using your Google Account.
2. If you don’t have one, create it first
3. Choose "Get started in console" option to use Firebase.

🧭 2. Setup Firebase Project

1. Go to https://console.firebase.google.com
2. Create a new project: name it "foobar".

3. Web Application

1. Add a web (Click the </> button).
2. Register an app by adding nickname: foo.
3. In the "Add Firebase SDK" page, choose `Use a <script> tag`
4. Copy the information so you can use it later.

- You can get this information at: left side bar -> Project Overview -> ⚙ -> Project settings -> General.

```
<script type="module">
  import { initializeApp } from ...
  import { getAnalytics } from ...
  const firebaseConfig = {
    apiKey: "...", // your API
    authDomain: ... // all the other components
  };

  // Initial⚙️ize Firebase
  const app = initializeApp(firebaseConfig);
  const analytics = getAnalytics(app);
</script>
```

5. Click "continue to console" button.

Choose "Cloud Firestore"

1. Click "Create database" button.
2. Choose "production mode" or "test mode" and click "create" button.

4. Make a new collection and add documents

- You can make a collection and add documents later using Dart application anytime later.
- But, you need to know how to make collection and data using the console.

What is a Collection?

- A collection is like a folder that holds documents
- Each document contains key-value pairs (like JSON)
- Collections are created automatically when you add the first document

Create a collection via Firebase Console

What is a document?

- A document is a JSON file.
- It corresponds to a record in SQL table.

Create fields via Firebase console

1. In the "Start a collection" dialog box, clcik "Auto-ID".
2. Add field "foo" and type "string"

- You can add more fileds later.

> Now, you have made this configuration:

- Your project name is: foobar
- You are using "firestore" database (you can add other features later)
- You choose to make "web application" and get firebase Config to access the database.
- You made a collection: `foo`.
- You created a simple document with a field `foo` (you can add other fields later too).

5. Change Rules

1. Go to Rules tab, and change the rule as follows:

   `allow read, write: if true;`

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if true; <-- change this line
    }
  }
}
```

6. Run the firebase_add_with_button.html

The file is in the "database/firebase/webapp" directory.

1. Open the directory with VSCode.
2. Run local web server with VSCode "Go Live" extension.
3. Open the HTML.
4. Click the "Add Document" button.

Notice that two lines of code should be added to use the web application.

```
import { getFirestore, collection, addDoc } from
  "https://www.gstatic.com/firebasejs/12.0.0/firebase-firestore.js";

const db = getFirestore(app);
```

7. Dart Aplication

Install Firebase CLI

install

```
curl -sL https://firebase.tools | bash

# Verify installation
firebase --version
```

login using your Google account

Check your existing firebase projects

Check your Project ID.

```
> firebase projects:list

✔ Preparing the list of your Firebase projects
```

Project Display Name	Project ID	Project Number	Resource Location ID
foobar	foobar-a1317	827133271343	[Not specified]

1 project(s) total.

Run foo application

1. Go to `database/firebase/foo`.
2. Open the main.dart file.
3. Give your project id to `Firestore.initialize`.

```
Future<void> main() async {
  // Initialize Firestore with your project ID (not Firebase app)
  Firestore.initialize("YOUR PROJECT ID"); // <---
```

4. Run `dart lib/main.dart` using command line.

```
Adding document...
Document added with ID: kYcgLeqH9MODettFExBk
Getting document data...
Retrieved data: {foo: xyz, bar: 75}
foo: xyz
bar: 75

Alternative - Get by ID:
Retrieved by ID: {foo: xyz, bar: 75}

Closing connection...
Program finished!
```

✅ You're Ready

- Firebase setup complete
- Firestore database connected
- Works for Web and Dart

🔷 Firebase Docs

(Optional) Running Firebase Web on Flutter

Use this information when you need to make Flutter applications that use Firebase.

[Firebase flutter setup]

Install and run the FlutterFire CLI

🟦 FlutterFire CLI

This is one time installation.

```
dart pub global activate flutterfire_cli
```

1. This is only for making Flutter application, so you should use it when you make Flutter app.
2. But for running dart application, you don't have to run these commands.

Flutter Web App

Use `database/firebase/foobar_flutter_webapp` as an example.

1. Create Flutter app using `fltter create`

```
mkdir foobar_flutter_webapp
cd foobar_flutter_webapp
flutter create . --platforms=web
```

2. Run `flutterfire configure`, choose your project and

4. Use the generated file in your application.

```dart
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'dart:math';
import 'firebase_options.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(MyApp());
}
```

5. Update `pubspec.yaml`

```
dependencies:
  flutter:
    sdk: flutter
  firebase_core: ^2.24.2
  cloud_firestore: ^4.13.6
  firebase_core_web: ^2.10.0
  cloud_firestore_web: ^3.8.10
```

6. Run `flutter pub get`.

7. Ensure your web/index.html includes Firebase SDK scripts: it's the same as the webapp example.

```html
<!DOCTYPE html>
<html>
<head>
  <!-- ... other head elements ... -->

  <!-- Firebase SDK -->
  <script src="https://www.gstatic.com/firebasejs/10.7.0/firebase-app-compat.js"></script>
  <script src="https://www.gstatic.com/firebasejs/10.7.0/firebase-firestore-compat.js"></script>
</head>
<body>
  <!-- ... body content ... -->

  <script>
    // Your web app's Firebase configuration
    const firebaseConfig = {
      // Your config here - get this from Firebase Console
    };

    // Initialize Firebase
    firebase.initializeApp(firebaseConfig);
  </script>
```

8. Use Firebase in the Flutter code.

```
final firestore = FirebaseFirestore.instance;

DocumentReference docRef = await firestore
    .collection(collectionName)
    .add(newData);
      print("Document added with ID: ${docRef.id}");

DocumentSnapshot docSnapshot = await docRef.get();

if (docSnapshot.exists) {
  Map<String, dynamic> savedData = docSnapshot.data() as Map<String, dynamic>;
  _currentFoo = savedData['foo'];
  _currentBar = savedData['bar'];
}
```

Adding mac/ios platforms

https://firebase.google.com/docs/flutter/setup?platform=ios

```
flutter create . --platforms=ios,macos
```

2. Run commands, choose your project and platforms.

```
> flutter pub add firebase_core
> flutter pub add cloud_firestore
> flutterfire configure
> flutter pub get
> flutter doctor # to check everything is OK
```

You don't need to add these packages when you already made flutter firebase/firestore app.

```
> flutter pub add firebase_core
> flutter pub add cloud_firestore
```

3. Update the main.dart

```dart
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'dart:math';
import 'firebase_options.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(const MyApp());
}
```

4. Checking configuration files

macosx/Podfile

```
platform :osx, '10.15'
```

open macos/Runner.xcworkspace

- Select the “Runner” target in the project navigator (left sidebar)
- Go to the General tab
- Find Deployment Info → macOS Deployment Target

To allow network, make sure `macos/Runner/Release.entitlements` file exists.

```
<!-- Network Access - Required for Firebase -->
<key>com.apple.security.network.client</key>
<true/>
 <!-- Network Server - Required for Firebase realtime features -->
<key>com.apple.security.network.server</key>
<true/>
```

5. Run `flutter run` and choose `macos`

```
flutter clean
flutter build macos
flutter run -d macos
```