

# **MAD & PWA**

## **Sem. VI Third Year Engineering**

### **2024-2025**

**Prof.Sujata Oak**  
**APSHAH Institute of Technology**  
**Assistant Professor (Department of IT)**  
**M.E. (Computer Engineering)**

**saoak@apsit.edu.in**

**9 8 1 9 0 3 0 9 7 5**

Firebase is

Google's app  
development platform

# Supported Platforms



iOS



Android



Web



C++



Unity



Flutter

# Firebase

- Firebase is a Backend-as-a-Service (BaaS) app development platform that provides hosted backend services .
- Such as a real-time database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files.

# Firebase

- Firebase has three main services, i.e., a real-time database, user authentication, and hosting.
- We can use these services with the help of the Firebase iOS SDK to create apps without writing any server code

 New Project X

Project name:

Project location:  ...

Description:

Project type:

Organization:

Android language:  Java  Kotlin

Platforms:  Android  iOS  Linux  MacOS  Web  Windows

When created, the new project will run on the selected platforms (others can be added later).

Create project offline

> More Settings

Previous Create Cancel



[firebase.google.com](https://firebase.google.com)

<https://firebase.google.com> › firebase

## Sign-In to Google Account

A screenshot of the Firebase website's navigation bar. The bar includes links for Build, Run, Solutions, Pricing, Docs, Community, Support, a search bar, language selection (English), and a sun icon. The "Go to console" button is highlighted with a thick red rectangular border.



Get started with a  
Firebase project

## x Create a project

Project name

**firebase series dummy**

[fir-series-dummy](#)

[apsit.org.in](#)

- I accept the [Firebase terms](#).
- I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.
  
- Join the [Google Developer Program](#) to enrich your developer journey with access to AI assistance, learning resources, profile badges, and more!

Already have a Google Cloud project?  
[Add Firebase to Google Cloud project](#)

**Continue**

- × Create a project

## Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:



A/B testing [?](#)



Breadcrumb logs in Crashlytics [?](#)



User segmentation & targeting across [?](#)  
Firebase products



Event-based Cloud Functions triggers [?](#)



Free unlimited reporting [?](#)



Enable Google Analytics for this project

Recommended

[Previous](#)

[Continue](#)

## Create a project

India 

Google Analytics is a business tool. Use it exclusively for purposes related to your trade, business, craft, or profession.

Data sharing settings and Google Analytics terms

- Use the default settings for sharing Google Analytics data. [Learn more](#)

- Share your Analytics data with Google to improve Google Products and Services
- Share your Analytics data with Google to enable Benchmarking
- Share your Analytics data with Google to enable Technical Support
- Share your Analytics data with Google Account Specialists

- I accept the [Google Analytics terms](#)

Upon project creation, a new Google Analytics property will be created and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#).

[Previous](#)

[Create project](#)



firebase series dummy

✓ Your Firebase project is ready

Continue



Project Overview



Generative AI

Build with Gemini

Genkit NEW

Product categories

Build

Run

Analytics

All products

Related development tools

[IDX](#) [Checks](#)

firebase series dummy ▾

# firebase series dum...

[Spark plan](#)[Getting started? Tell Gemini about your project](#)

## Get started by adding Firebase to your app

[Add an app to get started](#)

The screenshot shows the Android Studio interface with the following details:

- Project Bar:** Shows the project name "firebase\_series" and its location "D:\FLUTTER\firebase\_series".
- Toolbars:** Includes tabs for README.md, main.dart, android\build.gradle, and app\build.gradle.
- File Explorer:** Lists project files and folders:
  - firebase\_series (selected)
  - .dart\_tool
  - .idea
  - android [firebase\_series\_android]
    - app
      - src
    - build.gradle (selected)
    - gradle
    - .gitignore
    - build.gradle
    - firebase\_series\_android.iml
    - gradle.properties
    - gradlew
    - gradlew.bat
    - local.properties
    - settings.gradle
  - ios
  - lib
  - linux
  - macos
  - test
  - web
- Code Editor:** Displays the content of the app\build.gradle file. The code defines the android block with kotlinOptions and defaultConfig sections. It includes a TODO comment for specifying the application ID and provides default values for minSdk, targetSdk, versionCode, and versionName. Lines 23 through 32 are partially cut off at the bottom.

```
8     android {
17
18         kotlinOptions {
19             jvmTarget = JavaVersion.VERSION_1_8
20         }
21
22         defaultConfig {
23             // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id).
24             applicationId = "com.example.firebaseio_series"
25             // You can update the following values to match your application needs.
26             // For more information, see: https://flutter.dev/to/review-gradle-config
27             minSdk = flutter.minSdkVersion
28             targetSdk = flutter.targetSdkVersion
29             versionCode = flutter.versionCode
30             versionName = flutter.versionName
31
32 }
```

## 1 Register app

Android package name [?](#)

com.example.firebaseio\_series

App nickname (optional) [?](#)

## My Android App

Debug signing certificate SHA-1 (optional) [?](#)

 Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

## Register app

## 2 Download and then add config file

### 3 Add Firebase SDK

## 4 Next steps



## Register app

Android package name: com.example.firebaseio\_series

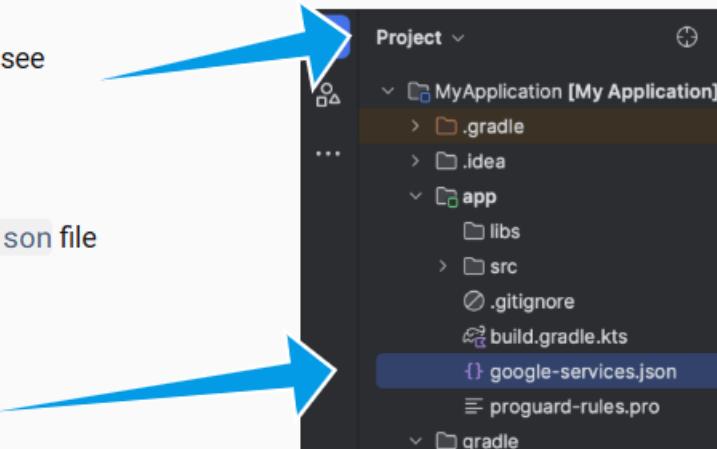


## Download and then add config file

Instructions for Android Studio below | [Unity](#) [C++](#)

[Download google-services.json](#)

Switch to the **Project** view in Android Studio to see your project root directory.



Move your downloaded `google-services.json` file into your module (app-level) root directory.



`google-services.json`

[Next](#)



## Add Firebase SDK



## Next steps

The screenshot shows the Android Studio interface with the following details:

- Project Tree:** On the left, the project structure is shown under the root folder "firebase\_series". Key files visible include ".dart\_tool", ".idea", "android", "gradle", ".gitignore", "build.gradle", "google-services.json", "gradle.properties", "gradlew", "gradlew.bat", "local.properties", "settings.gradle", "ios", "lib", "linux", and "macos".
- File Tab:** At the top, the "app\build.gradle" tab is selected.
- Code Editor:** The main area displays the content of the build.gradle file. The code defines an "android" block with "kotlinOptions" and "defaultConfig" sections. The "applicationId" is set to "com.example.firebaseio\_series".

```
8 android {
17
18     kotlinOptions {
19         jvmTarget = JavaVersion.VERSION_1_8
20     }
21
22     defaultConfig {
23         // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id).
24         applicationId = "com.example.firebaseio_series"
25         // You can update the following values to match your application needs.
26         // For more information, see: https://flutter.dev/to/review-gradle-configuration
27         minSdk = flutter.minSdkVersion
28         targetSdk = flutter.targetSdkVersion
29         versionCode = flutter.versionCode
30         versionName = flutter.versionName
31 }
```



## Register app

Android package name: com.example.firebaseio\_series



## Download and then add config file



## Add Firebase SDK

Instructions for Gradle | [Unity](#) [C++](#)



Are you still using the `buildscript syntax` to manage plugins? Learn how to [add Firebase plugins](#) using that syntax.

1. To make the `google-services.json` config values accessible to Firebase SDKs, you need the Google services Gradle plugin.



Kotlin DSL (`build.gradle.kts`)



Groovy (`build.gradle`)

Add the plugin as a dependency to your **project-level** `build.gradle.kts` file:

**Root-level (project-level) Gradle file** (`<project>/build.gradle.kts`):

```
plugins {
    // ...

    // Add the dependency for the Google services Gradle plugin
    id("com.google.gms.google-services") version "4.4.2" apply false
}
```

The screenshot shows the Android Studio interface with the project 'firebase\_series' open. The left sidebar displays the project structure, including a .dart\_tool folder, an .idea folder, an android folder containing an app folder with a build.gradle file, a gradle folder, a .gitignore file, and two build.gradle files (one in the root and one in the android/app directory). The top navigation bar has tabs for README.md, main.dart, android\build.gradle, and app\build.gradle, with the app\build.gradle tab currently selected. The code editor on the right contains the following content:

```
1  plugins {  
2      id "com.android.application"  
3      id "kotlin-android"  
4      // The Flutter Gradle Plugin must be applied after the Android and Kotlin  
5      id "dev.flutter.flutter-gradle-plugin"  
6      id 'com.google.gms.google-services'  
7  }  
8 }  
9 }
```

The screenshot shows the Android Studio interface with the following details:

- Project View:** On the left, the project structure is displayed under the root folder "firebase\_series". Key files shown include ".dart\_tool", ".idea", "android [firebase\_series\_android]", "app", "src", "build.gradle", "google-services.json", "gradle", ".gitignore", "build.gradle", "firebase\_series\_android.iml", "gradle.properties", "gradlew", "gradlew.bat", "local.properties", "settings.gradle", "build", "ios", "lib", "linux", and "macos".
- Editor Tab Bar:** At the top, tabs for "README.md", "main.dart", "android\build.gradle", and "app\build.gradle" are visible. The "app\build.gradle" tab is currently selected.
- Code Editor Content:** The main editor area contains the following code:

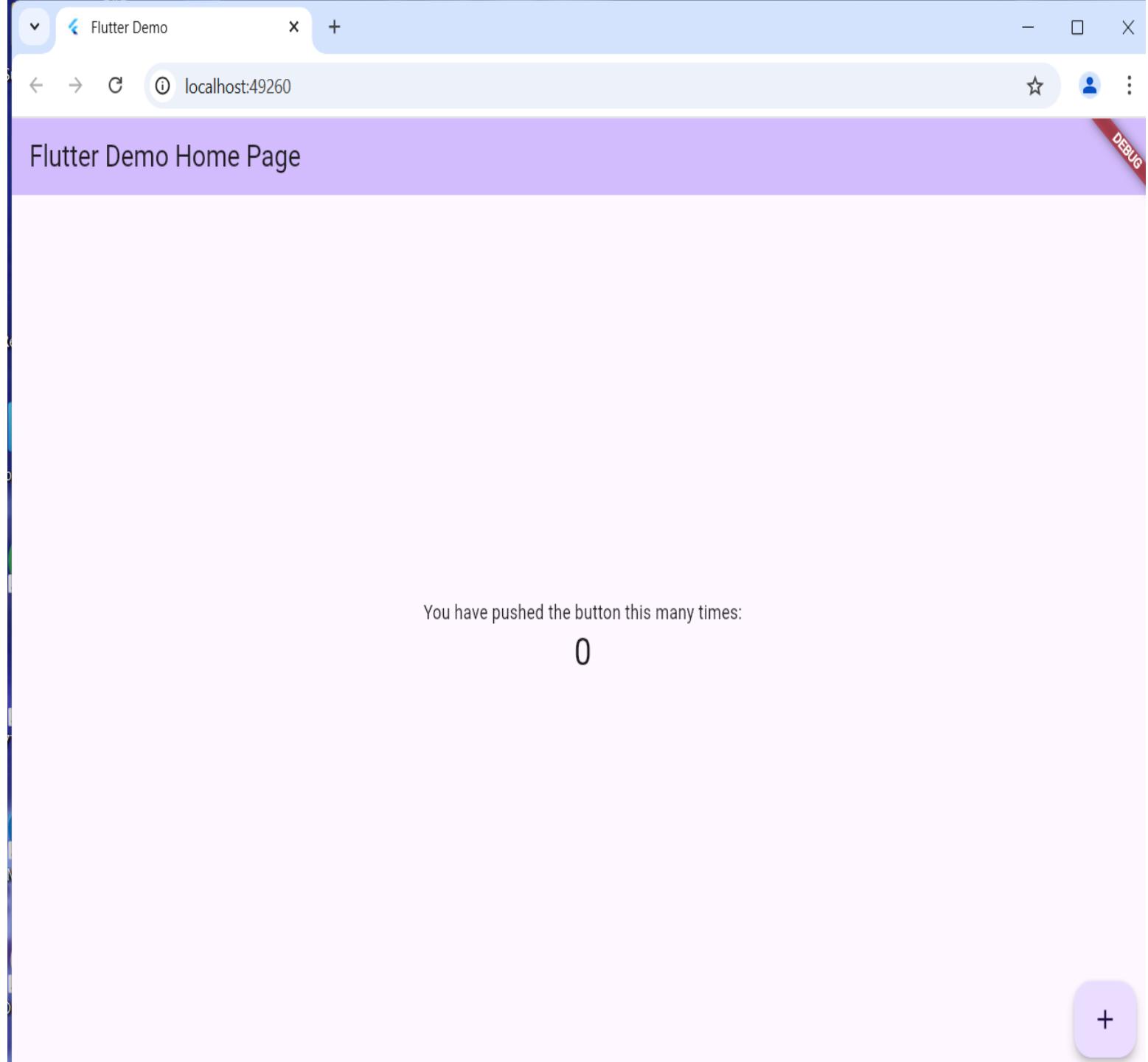
```
flutter {
    source = "../.."
}

dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:33.10.0')

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation 'com.google.firebase.firebaseio'

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

# RUN Project





Sign in Help

firebase core

Platforms

- Android
- iOS
- Linux
- macOS
- Web
- Windows

RESULTS 254 packages

SORT BY SEARCH RELEVANCE

Matching package name: **firebase**

**firebase\_core**

3.79k | 160 | 2.64M  
LIKES POINTS DOWNLOADS

Flutter plugin for Firebase Core, enabling connecting to multiple Firebase apps. #firebase #core

v 3.12.1 (5 days ago) firebase.google.com Flutter Favorite Dart 3 compatible

SDKs

SDK FLUTTER PLATFORM ANDROID IOS MACOS WEB WINDOWS

License

API result: [firebase\\_core/](#)

Advanced

**flutter\_chat\_types**

47 | 160 | 66.5k  
LIKES POINTS DOWNLOADS

Utility library for the flutter\_chat\_ui and flutter\_firebase\_chat\_core libraries which contains shared type declarations.

The screenshot shows the Android Studio interface with the project structure on the left and the pubspec.yaml file content on the right.

**Project Structure:**

- gradlew.bat
- local.properties
- settings.gradle
- > build (highlighted)
- > ios
- > lib
- > linux
- > macos
- > test (highlighted)
- > web
- > windows
- .gitignore
- .metadata
- analysis\_options.yaml
- firebase\_series.iml
- pubspec.lock
- pubspec.yaml (highlighted)
- README.md
- > External Libraries

**pubspec.yaml Content:**

```
Flutter commands
Pub get Pub upgrade Pub outdated Flutter doc

# the latest version available on pub.dev. To see which dependencies have been updated, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8
  firebase_core: ^3.12.1

dev_dependencies:
  flutter_test:
    sdk: flutter
```



firebase auth

Platforms

- Android
- iOS
- Linux
- macOS
- Web
- Windows

RESULTS 336 packages

SORT BY SEARCH RELEVANCE

Matching package names: [firebase](#), [auth](#)

[firebase\\_auth](#)

4.08k | 160 | 903k  
LIKES POINTS DOWNLOADS

Flutter plugin for Firebase Auth, enabling authentication using passwords, phone numbers and identity providers like Google, Facebook and Twitter. [#firebase](#) [#authentication](#) [#identity](#) [#sign-in](#) [#sign-up](#)

SDKs

v 5.5.1 (5.days ago) [firebase.google.com](#) [Flutter Favorite](#) [Dart 3 compatible](#)

License

[SDK](#) [FLUTTER](#) [PLATFORM](#) [ANDROID](#) [IOS](#) [MACOS](#) [WEB](#) [WINDOWS](#)



cloud firestore

## Platforms

- Android
- iOS
- Linux
- macOS
- Web
- Windows

RESULTS 153 packages

SORT BY SEARCH RELEVANCE

[cloud\\_firestore](#)

3.62k | 160 | 797k

LIKES POINTS DOWNLOADS

Flutter plugin for Cloud Firestore, a cloud-hosted, noSQL database with live synchronization and offline support on Android and iOS. #firebase #firestore #realtime #database

v 5.6.5 (5 days ago) firebase.google.com BSD-3-Clause Flutter Favorite Dart 3 compatible

[SDK](#) [FLUTTER](#) [PLATFORM](#) [ANDROID](#) [IOS](#) [MACOS](#) [WEB](#) [WINDOWS](#)

API result: [cloud\\_firestore/](#)

firebase storage

## Platforms

- Android
- iOS
- Linux
- macOS
- Web
- Windows

RESULTS 252 packages

SORT BY SEARCH RELEVANCE

Matching package names: **firebase, storage****firebase\_storage** **1.69k** LIKES | **150** POINTS | **440k** DOWLOADS

Flutter plugin for Firebase Cloud Storage, a powerful, simple, and cost-effective object storage service for Android and iOS. **#firebase #storage #upload #download #files**

The screenshot shows the Android Studio interface with the pubspec.yaml file open in the main editor area. The left sidebar displays the project structure, and the top navigation bar includes tabs for README.md, main.dart, android\build.gradle, app\build.gradle, widget\_test.dart, and pubspec.yaml.

**Flutter commands**

Pub get    Pub upgrade    Pub outdated    Flutter doc

```
28 # the latest version available on pub.dev. To see which dependencies have
29 # versions available, run `flutter pub outdated`.
30 dependencies:
31   flutter:
32     sdk: flutter
33
34   # The following adds the Cupertino Icons font to your application.
35   # Use with the CupertinoIcons class for iOS style icons.
36   cupertino_icons: ^1.0.8
37   firebase_core: ^3.12.1
38   firebase_auth: ^5.5.1
39   cloud_firestore: ^5.6.5
40   firebase_storage: ^12.4.4
41 dev_dependencies:
```

## **5.1. Install Firebase CLI**

**npm install -g firebase-tools**

## **5.2. Login to Firebase**

**firebase login**

## **5.3. Initialize Firebase in the Project**

**firebase init**

- Select **Hosting**.
- Choose "**Use an existing project**" and select counter\_app.
- Set the public directory to build/web.

## **5.4. Build the Flutter Web App**

**flutter build web**

## **5.5. Deploy the App**

**firebase deploy**

```
PS D:\FLUTTER\firebase_series> npm install -g firebase-tools
```

```
added 631 packages in 1m
```

```
71 packages are looking for funding
```

```
  run `npm fund` for details
```

```
npm notice
```

```
npm notice New major version of npm available! 10.8.2 -> 11.1.0
```

```
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.1.0
```

```
npm notice To update run: npm install -g npm@11.1.0
```

```
npm notice
```

```
PS D:\FLUTTER\firebase_series> █
```

```
PS D:\FLUTTER\firebase_series> firebase login
```

i Firebase optionally collects CLI and Emulator Suite usage and error reporting information to help improve our products. Data is collected in accordance with Google's privacy policy (<https://policies.google.com/privacy>) and is not used to identify you.

? Allow Firebase to collect CLI and Emulator Suite usage and error reporting information? Yes

i To change your data collection preference at any time, run `firebase logout` and log in again.

Visit this URL on this device to log in:

[https://accounts.google.com/o/oauth2/auth?client\\_id=563584335869-fgrhgmd47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response\\_type=code&state=382818818&redirect\\_uri=http%3A%2F%2Flocalhost%3A9005](https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhgmd47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response_type=code&state=382818818&redirect_uri=http%3A%2F%2Flocalhost%3A9005)

Waiting for authentication...

+ Success! Logged in as **saoak@apsit.edu.in**

```
PS D:\FLUTTER\firebase_series> firebase init
```

You're about to initialize a Firebase project in this directory:

D:\FLUTTER\firebase\_series

? Are you ready to proceed? (Y/n)

D:\FLUTTER\firebase\_series

? Are you ready to proceed? Yes

? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)

- ( ) Genkit: Setup a new Genkit project with Firebase
- ( ) Functions: Configure a Cloud Functions directory and its files
- ( ) App Hosting: Configure an apphosting.yaml file for App Hosting
- >(\* Host: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
- ( ) Storage: Configure a security rules file for Cloud Storage
- ( ) Emulators: Set up local emulators for Firebase products
- ( ) Remote Config: Configure a template file for Remote Config

- ? Please select an option: (Use arrow keys)
- > Use an existing project
  - Create a new project
  - Add Firebase to an existing Google Cloud Platform project
  - Don't set up a default project
-

### ==== Hosting Setup

Your **public** directory is the folder (relative to your project directory) that will contain Hosting assets to be uploaded with **firebase deploy**. If you have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? build/web

```
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
+ Wrote build/web/index.html

i  Writing configuration info to firebase.json...
i  Writing project information to .firebaserc...

+ Firebase initialization complete!
```

<https://www.youtube.com/watch?v=VqT1utkYIfY>

## A New Project

X

Project name: finalfirebaseproject

Project location: D:\FLUTTER\finalfirebaseproject

Description: A new Flutter project.

Project type: Application

Organization: com.example

Android language:  Java  Kotlin

Platforms:  Android  iOS  Linux  MacOS  Web  Windows

When created, the new project will run on the selected platforms (others can be added later).

Create project offline

> More Settings



firebase console

All

Images

Videos

Shopping

Short videos

Forums

News

More



Firebase Console

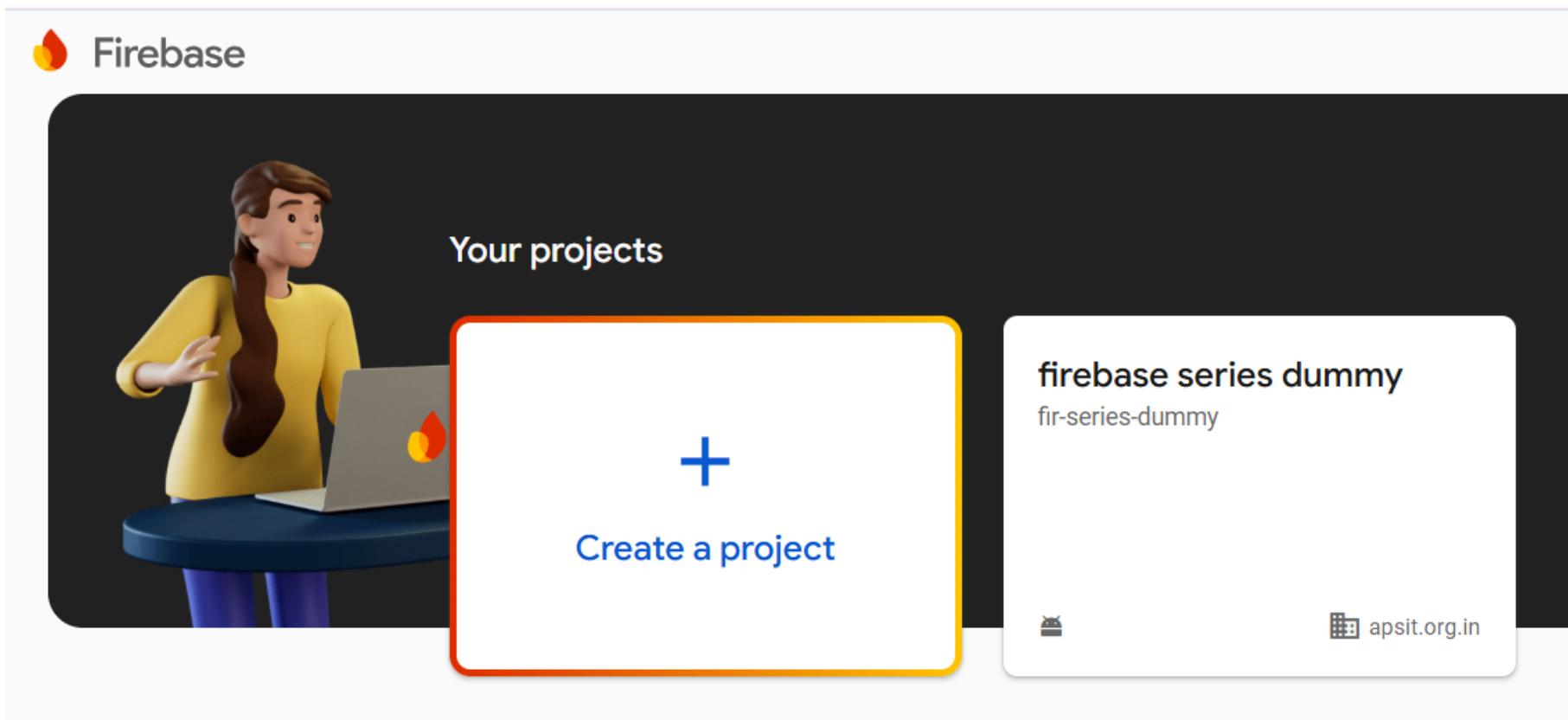
<https://console.firebaseio.google.com>

⋮

[Firebase Console - Google](#)

Not your computer? Use Guest mode to sign in privately. Learn more about using Guest mode. Next.

Create account. For my personal use; For work or my ...



x Create a project

Let's start with a name for  
your project <sup>?</sup>

Project name

firebase-setup

 fir-setup-a38db

 apsit.org.in

Already have a Google Cloud project?  
[Add Firebase to Google Cloud project](#)

Continue

- × Create a project

## Google Analytics

### for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:

×

A/B testing ?

×

Breadcrumb logs in Crashlytics ?

×

User segmentation & targeting across ?  
Firebase products

×

Event-based Cloud Functions triggers ?

×

Free unlimited reporting ?



Enable Google Analytics for this project  
Recommended



firebase-setup



Your Firebase project is ready

Continue



Project Overview



Generative AI

Build with Gemini

Genkit NEW

Product categories

Build



Run



Analytics



All products

Related development tools

[IDX](#) [Checks](#)

firebase-setup ▾

firebase-setup

plan

Getting started? Tell Gemini about your project

# Get started by adding Firebase to your app



Add an app to get started



Click on Android

## D:\FLUTTER\finalfirebaseproject\android\app\build.gradle

The screenshot shows the Android Studio interface with the project navigation bar at the top. The 'Project' tab is selected, showing a tree view of the project structure. Under the 'finalfirebaseproject' node, the 'build.gradle' file is highlighted and selected. The main editor area displays the content of the build.gradle file for the android/app module.

```
8     android {
20
21
22     defaultConfig {
23         // TODO: Specify your own unique Application ID (https://developer.android.c
24         applicationId = "com.example.finalfirebaseproject"
25         // You can update the following values to match your application needs.
26         // For more information, see: https://flutter.dev/to/review-gradle-config.
27         minSdk = flutter.minSdkVersion
28         targetSdk = flutter.targetSdkVersion
29         versionCode = flutter.versionCode
30         versionName = flutter.versionName
31     }
}
```

## **× Add Firebase to your Android app**

## 1 Register app

Android package name [?](#)

com.example.finalfirebaseproject

App nickname (optional) [?](#)

# My Android App

Debug signing certificate SHA-1 (optional) [?](#)

**(i)** Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

## Register app

## Add Firebase to your Android app

### Register app

Android package name: com.example.finalfirebaseproject

### 2 Download and then add config file

Instructions for Android Studio below | [Unity](#) | [C++](#)

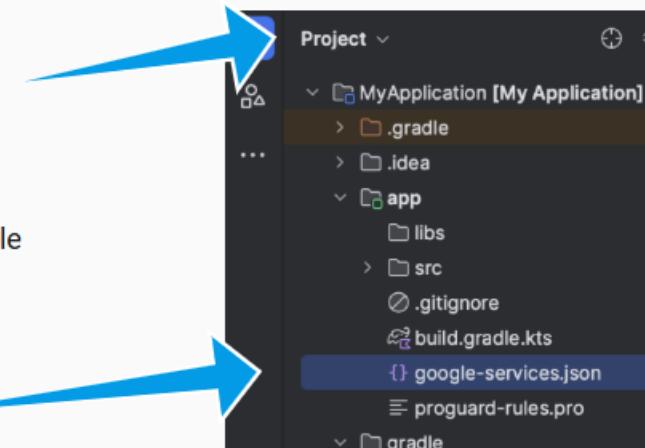
 [Download google-services.json](#)

Switch to the **Project** view in Android Studio to see your project root directory.

Move your downloaded `google-services.json` file into your module (app-level) root directory.



`google-services.json`



[Next](#)

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right.

**Project Structure:**

- finalfirebaseproject (selected)
- .dart\_tool
- .idea
- android [finalfirebaseproject]
  - app (selected)
  - src
  - build.gradle
  - google-services.json
- gradle
- .gitignore
- build.gradle
- finalfirebaseproject\_android
- gradle.properties
- gradlew
- gradlew.bat
- local.properties
- settings.gradle
- ios
- lib
- linux
- macos
- test
- web
- windows
- .gitignore
- .metadata
- analysis\_options.yaml

**Code Editor (google-services.json):**

```
1  {
2      "project_info": {
3          "project_number": "858467345095",
4          "project_id": "fir-setup-a38db",
5          "storage_bucket": "fir-setup-a38db.firebaseio.storage.app"
6      },
7      "client": [
8          {
9              "client_info": {
10                  "mobilesdk_app_id": "1:858467345095:android:bfd1e11f801cb7c1a6d4e2",
11                  "android_client_info": {
12                      "package_name": "com.example.finalfirebaseproject"
13                  }
14              },
15              "oauth_client": [],
16              "api_key": [
17                  {
18                      "current_key": "AIzaSyC-fZYUe1JIZDBYYU2op07Z224h8_Ynuek"
19                  }
20              ],
21              "services": {

```

### 3 Add Firebase SDK

Instructions for Gradle | [Unity](#) [C++](#)

★ Are you still using the `buildscript` syntax to manage plugins? Learn how to [add Firebase plugins](#) using that syntax.

1. To make the `google-services.json` config values accessible to Firebase SDKs, you need the Google services Gradle plugin.

- Kotlin DSL (`build.gradle.kts`)     Groovy (`build.gradle`)

Add the plugin as a dependency to your **project-level** `build.gradle` file:

**Root-level (project-level) Gradle file** (`<project>/build.gradle`):

```
plugins {  
    // ...  
  
    // Add the dependency for the Google services Gradle plugin  
    id 'com.google.gms.google-services' version '4.4.2' apply false  
}
```



The screenshot shows the Android Studio interface with the project structure on the left and the build configuration file on the right.

**Project Structure:**

- finalfirebaseproject (selected)
- .dart\_tool
- .idea
- android [finalfirebaseproject]

  - app
  - gradle
  - .gitignore

- build.gradle (selected)
- finalfirebaseproject\_android
- gradle.properties
- gradlew
- gradlew.bat
- local.properties
- settings.gradle
- ios
- lib
- linux
- macos
- test (selected)
- web
- windows
- .gitignore

**build.gradle Content:**

```
subprojects {  
    project.buildDir = "${rootProject.buildDir}/${project.name}"  
}  
  
subprojects {  
    project.evaluationDependsOn(":app")  
}  
  
tasks.register("clean", Delete) {  
    delete rootProject.buildDir  
}  
  
dependencies{  
    // id 'com.google.gms.google-services' version '4.4.2' apply false  
    classpath 'com.google.gms.google-services:4.4.2'  
}
```

2. Then, in your **module (app-level)** `build.gradle` file, add both the `google-services` plugin and any Firebase SDKs that you want to use in your app:

**Module (app-level) Gradle file (`<project>/<app-module>/build.gradle`):**

```
plugins {  
    id 'com.android.application'  
    // Add the Google services Gradle plugin  
    id 'com.google.gms.google-services'  
    ...  
}  
  
dependencies {  
    // Import the Firebase BoM  
    implementation platform('com.google.firebase:firebase-bom:33.10.0')  
  
    // TODO: Add the dependencies for Firebase products you want to use  
    // When using the BoM, don't specify versions in Firebase dependencies  
    // https://firebase.google.com/docs/android/setup#available-libraries  
}
```

The screenshot shows the Android Studio interface with the project navigation bar at the top. The current file is `app\build.gradle`, which contains the following code:

```
plugins {
    id "com.android.application"
    id "kotlin-android"
    // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle Plugins
    id "dev.flutter.flutter-gradle-plugin"
    id 'com.google.gms.google-services'
}

android {
    namespace = "com.example.finalfirebaseproject"
    compileSdk = flutter.compileSdkVersion
    ndkVersion = flutter.ndkVersion
}
```

The code editor highlights several lines with green syntax coloring, indicating they are part of the build configuration. The file path `D:\FLUTTER\finalfirebaseproject\android\app\build.gradle` is visible in the top right corner of the code editor.

2. Then, in your **module (app-level)** `build.gradle` file, add both the `google-services` plugin and any Firebase SDKs that you want to use in your app:

**Module (app-level) Gradle file (`<project>/<app-module>/build.gradle`):**

```
plugins {  
    id 'com.android.application'  
    // Add the Google services Gradle plugin  
    id 'com.google.gms.google-services'  
    ...  
}  
  
dependencies {  
    // Import the Firebase BoM  
    implementation platform('com.google.firebase:firebase-bom:33.10.0')  
  
    // TODO: Add the dependencies for Firebase products you want to use  
    // When using the BoM, don't specify versions in Firebase dependencies  
    // https://firebase.google.com/docs/android/setup#available-libraries  
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

The screenshot shows the Android Studio interface with the project navigation bar on the left and the code editor on the right.

**Project Tree:**

- finalfirebaseproject D:\FLUTTER
- .dart\_tool
- .idea
- android [finalfirebaseproject]
- app
- src
- build.gradle
- google-services.json
- gradle
- .gitignore
- build.gradle
- finalfirebaseproject\_android.
- gradle.properties
- gradlew
- gradlew.bat
- local.properties
- settings.gradle
- ios
- lib
- linux
- macos
- test
- web
- windows
- .gitignore

**Code Editor (app/build.gradle):**

```
9     android {
10         buildTypes {
11             ...
12         }
13     }
14
15     flutter {
16         source = "..."|
17     }
18
19     dependencies {
20         // Import the Firebase BoM
21         implementation platform('com.google.firebaseio:firebase-bom:33.10.0')
22
23         ...
24         // TODO: Add the dependencies for Firebase products you want to use
25         // When using the BoM, don't specify versions in Firebase dependencies
26         // https://firebase.google.com/docs/android/setup#available-libraries
27     }
28 }
```

Click → Next Now Android App is configured for firebase project

3. After adding the plugin and the desired SDKs, sync your Android project with Gradle files.

[Previous](#)

[Next](#)

## × Add Firebase to your Android app

### Register app

Android package name: com.example.finalfirebaseproject

### Download and then add config file

### Add Firebase SDK

### 4 Next steps

You're all set!

Make sure to check out 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 console



Firebase

Project Overview



Generative AI

Build with Gemini

Genkit NEW

Product categories

Build ▼

Run ▼

Analytics ▼

All products

Related development tools

firebase-setup ▾

# firebase-setup

Spark plan

Getting started? Tell Gemini about your project



com.example.final...

Add app

## Share your feedback with Firebase

Participate in a Firebase research study to help us drive Firebase's full potential for all users!

[Participate now](#)



## Choose a product to add to your app

[Accelerate app development](#)

X

X

Click on Add app and configure for web

## Add Firebase to your web app

### 1 Register app

App nickname 

mywebfirebase

Also set up **Firebase Hosting** for this app. [Learn more](#) 

Hosting can also be set up later. There is no cost to get started anytime.



fir-setup-a38db (No deploys yet) 

**Register app**



pub dev

X

All

Images

Videos

Shopping

Short videos

Forums

News

More



Dart packages

<https://pub.dev>

⋮

## Pub.dev

A string-based path manipulation library. All of the path operations you know and love, with solid support for Windows, POSIX (Linux and Mac OS X), and the web.

The screenshot shows the pub.dev website interface. At the top left is the logo. A search bar contains the query "firebase core". Below the search bar, there are navigation icons and a progress bar. On the left, a sidebar titled "Platforms" lists checkboxes for "Android", "iOS", "Linux", and "macOS". To the right of the sidebar, the word "RESULTS" is followed by "254 packages". A vertical blue line highlights the search results. The first result listed is "Matching package name: [firebase](#)". Below it is the second result, "firebase\_core", also with a blue link and a small icon.

Platforms

RESULTS 254 packages

Android

iOS

Linux

macOS

Matching package name: [firebase](#)

[firebase\\_core](#)

Project ▾

M README.md main.dart app\build.gradle pubspec.yaml android\build.gradle ⌂ ⌂

Flutter commands

Pub get Pub upgrade

```
28 # the latest version available on pub.dev. To see which depends
29 # versions available, run `flutter pub outdated`.
30 dependencies:
31   flutter:
32     sdk: flutter
33
34   # The following adds the Cupertino Icons font to your application.
35   # Use with the CupertinoIcons class for iOS style icons.
36   cupertino_icons: ^1.0.8
37   firebase_core: ^3.12.1
```

## 2 Add Firebase SDK

- Use npm     Use a <script> tag

If you're already using [npm](#) and a module bundler such as [webpack](#) or [Rollup](#), you can run the following command to install the latest SDK ([Learn more](#)):

```
$ npm install firebase
```

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

[Goto main.dart](#)

```
// 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: "AIzaSyB5wuVRA26lHZ6fKUkdMEEMaBguuiCFOiw",
  authDomain: "fir-setup-a38db.firebaseio.com",
  projectId: "fir-setup-a38db",
  storageBucket: "fir-setup-a38db.firebaseio.storage.app",
  messagingSenderId: "858467345095",
  appId: "1:858467345095:web:50e7fd404f51efd7a6d4e2"
};

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

The screenshot shows the Android Studio interface with the main.dart file open in the editor. The project structure on the left includes files like README.md, app\build.gradle, pubspec.yaml, android\build.gradle, and google-services.json. The main.dart file contains Dart code for initializing Firebase:

```
5 void main() {  
6     if(kIsWeb){  
7         Firebase.initializeApp(  
8             options:const FirebaseOptions(  
9                 apiKey: "AIzaSyB5wuVRA26lHZ6fKUkdMEEMaBguuiCF0iw",  
10                authDomain: "fir-setup-a38db.firebaseio.com",  
11                projectId: "fir-setup-a38db",  
12                storageBucket: "fir-setup-a38db.firebaseiostorage.app",  
13                messagingSenderId: "858467345095",  
14                appId: "1:858467345095:web:50e7fd404f51efd7a6d4e2"));  
15 }else{  
16     Firebase.initializeApp();  
17 }  
18  
19 runApp(const MyApp());  
20 }  
21 }
```

**Next**

The screenshot shows the Android Studio interface with the main.dart file open in the editor. The code initializes Firebase with specific options based on the environment (kIsWeb). It includes details like apiKey, authDomain, projectId, storageBucket, messagingSenderId, and appId.

```
4
5 void main() async {
6     WidgetsFlutterBinding.ensureInitialized();
7     if(kIsWeb){
8         await Firebase.initializeApp(
9             options:const FirebaseOptions(
10                 apiKey: "AIzaSyB5wuVRA26lHZ6fKUkdMEEMaBguuiCF0iw",
11                 authDomain: "fir-setup-a38db.firebaseio.com",
12                 projectId: "fir-setup-a38db",
13                 storageBucket: "fir-setup-a38db.firebaseiostorage.app",
14                 messagingSenderId: "858467345095",
15                 appId: "1:858467345095:web:50e7fd404f51efd7a6d4e2"));
16     }else{
17         await Firebase.initializeApp();
18     }
19
20
21 runApp(const MyApp());
```

## × Add Firebase to your web app

✓ Register app

>Add Firebase SDK

3 Install Firebase CLI

To host your site with Firebase Hosting, you need the Firebase CLI (a command line tool).

Run the following [npm](#) command to install the CLI or update to the latest CLI version.

```
$ npm install -g firebase-tools
```



Doesn't work? Take a look at the [Firebase CLI reference](#) or change your [npm permissions](#).

Previous

Next

```
PS D:\FLUTTER\finalfirebaseproject> npm install -g firebase-tools  
changed 631 packages in 37s  
  
71 packages are looking for funding  
  run `npm fund` for details
```

```
PS D:\FLUTTER\finalfirebaseproject> firebase login  
Already logged in as saoak@apsit.edu.in
```

## 4 Deploy to Firebase Hosting

You can deploy now or [later ↗](#). To deploy now, open a terminal window, then navigate to or create a root directory for your web app.

[Sign in to Google](#)

```
$ firebase login
```



**Initiate your project**

Run this command from your app's root directory:

```
$ firebase init
```



**When you're ready, deploy your web app**

Put your static files (e.g., HTML, CSS, JS) in your app's deploy directory (the default is "public"). Then, run this command from your app's root directory:

```
$ firebase deploy
```



After deploying, view your app at [fir-setup-a38db.web.app ↗](#)

Need help? Check out the [Hosting docs ↗](#)

[Previous](#)

**Continue to console**

```
PS D:\FLUTTER\finalfirebaseproject> firebase init
```

```
##### ## ##### ##### ##### ##### ##### ##### ##### ##### #####  
## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##  
##### ## ##### ##### ##### ##### ##### ##### ##### ##### #####  
## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##  
## ##### ## ## ## ##### ##### ##### ##### ##### ##### #####
```

You're about to initialize a Firebase project in this directory:

D:\FLUTTER\finalfirebaseproject

? Are you ready to proceed? (Y/n)

You're about to initialize a Firebase project in this directory:

D:\FLUTTER\finalfirebaseproject

```
? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to
confirm your choices. (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
( ) Genkit: Setup a new Genkit project with Firebase
( ) Functions: Configure a Cloud Functions directory and its files
( ) App Hosting: Configure an apphosting.yaml file for App Hosting
>(*) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
( ) Storage: Configure a security rules file for Cloud Storage
( ) Emulators: Set up local emulators for Firebase products
( ) Remote Config: Configure a template file for Remote Config
(Move up and down to reveal more choices)
```

## ==== Project Setup

First, let's associate this project directory with a Firebase project. You can create multiple project aliases by running `firebase use --add`, but for now we'll just set up a default project.

? Please select an option: (Use arrow keys)

> Use an existing project

Create a new project

Add Firebase to an existing Google Cloud Platform project

Don't set up a default project

? Please select an option: Use an existing project

? Select a default Firebase project for this directory:

fir-series-dummy (firebase series dummy)

> fir-setup-a38db (firebase-setup)

Your `public` directory is the folder (relative to your project directory) that will contain Hosting assets to be uploaded with `firebase deploy`. If you have a build process for your assets, use your build's output directory.

- ? What do you want to use as your public directory? `build/web`
- ? Configure as a single-page app (rewrite all urls to `/index.html`)? `Yes`
- ? Set up automatic builds and deploys with GitHub? `No`
- + Wrote `build/web/index.html`
  
- i Writing configuration info to `firebase.json...`
- i Writing project information to `.firebaserc...`
  
- + Firebase initialization complete!

```
PS D:\FLUTTER\finalfirebaseproject> firebase deploy
```

```
==== Deploying to 'fir-setup-a38db'...
```

```
i  deploying hosting
i  hosting[fir-setup-a38db]: beginning deploy...
i  hosting[fir-setup-a38db]: found 1 files in build/web
+  hosting[fir-setup-a38db]: file upload complete
i  hosting[fir-setup-a38db]: finalizing version...
+  hosting[fir-setup-a38db]: version finalized
i  hosting[fir-setup-a38db]: releasing new version...
+  hosting[fir-setup-a38db]: release complete

+  Deploy complete!
```

Project Console: <https://console.firebaseio.google.com/project/fir-setup-a38db/overview>

Hosting URL: <https://fir-setup-a38db.web.app>





