



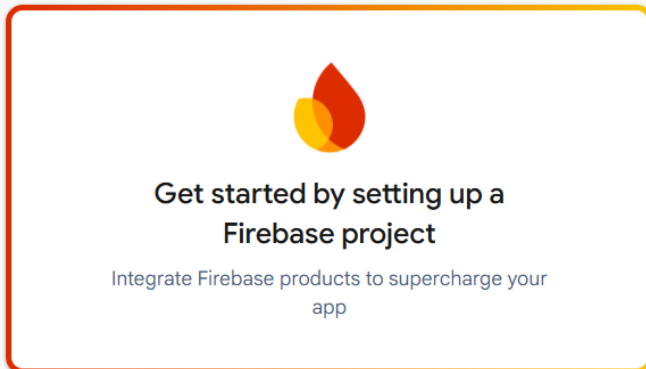
This guide explains how to create a free **Firebase** project — required to activate the OTA Guard mobile companion app (real-time alerts, price control from your phone). Google does not ask for **any bank card** for these steps. Allow about 10 minutes.

1

Create the Firebase project

1. Go to <https://console.firebase.google.com/>, signed in with the hotel's Google account.

Get started



Landing screen — click the "Get started by setting up a Firebase project" card

2. Click the **"Get started by setting up a Firebase project"** card.
3. Enter a project name — Firebase automatically generates a technical ID below it based on this name (editable if needed).
4. Check **"I accept the Firebase terms"** (required).
5. Leave the **"Join the Google Developer Programme"** toggle off (not needed).
6. Click **"Continue"**.

× Create a project

Let's start with a name for your project[®]

Project name

OTA GUARD

ota-guard

☒ I accept the [Firebase terms](#)

☐ Join the [Google Developer Programme](#) 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

Project naming and Firebase terms confirmation

7. On the **"AI assistance for your Firebase project"** screen (Gemini in Firebase), leave the **"Enable Gemini in Firebase"** toggle off — not needed for OTA Guard, even though Google marks it "Recommended" — then click **"Continue"**.

×

Create a project

AI assistance for your Firebase project

Gemini in Firebase is integrated within the Firebase console to help streamline your development process.

- ◆ Chat with Gemini to plan and design your application, troubleshoot issues and get recommendations tailored to your project
- 🔧 Debug and troubleshoot issues in your apps using AI assistance in Firebase Crashlytics
- 📧 Get campaign insights and recommendations to engage your users through Firebase Cloud Messaging
- 🔗 Generate schema and explore your data using natural language in Firebase SQL Connect

Enable Gemini in Firebase

Recommended

Disclaimers:
Gemini in Firebase is still under development and may give inaccurate information. Test any code that it generates before using it in your apps. Gemini can interact with your Firebase data and tailor responses to your app, and it may use your prompts or data for training. [Learn more about how we train our models](#) Use of Gemini in Firebase is subject to [Google Terms of Service](#) and the [Generative AI Prohibited Use Policy](#)

Previous

Continue

Gemini in Firebase prompt — leave disabled

- On the **"Google Analytics for your Firebase project"** screen, leave the **"Enable Google Analytics for this project"** toggle off (not needed here).
- Click **"Create project"** and wait about 30 seconds.

×

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-and-targeting across Firebase products ⓘ

×

Event-based-Cloud-Functions-triggers ⓘ

×

Free-unlimited-reporting ⓘ

Enable Google Analytics for this project

Recommended

Previous

Create project

Google Analytics prompt — leave disabled

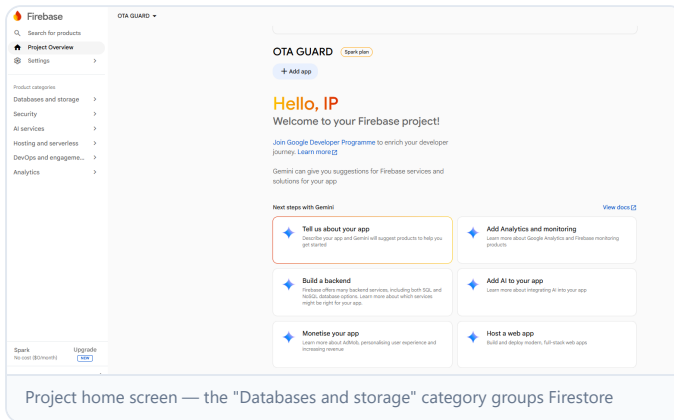
EXAMPLE

Suggested project name: `ota-guard-jangala` (fixed prefix + hotel name, lowercase, no spaces)

2

Enable Firestore Database

1. Once the project is created, you land on the project home screen ("Hello... Welcome to your Firebase project").
2. In the left menu, under "**Product categories**", click "**Databases and storage**" to expand that category (it is collapsed by default — Firestore does not appear directly in the menu).



Project home screen — the "Databases and storage" category groups Firestore

3. Click "**Firestore**" (under the "**NoSQL**" section), which appears in the expanded list.
4. Click "**Create database**".
5. Choose a region close to the hotel (e.g. `asia-south1` for India, `eur3` for Europe).
6. Select "**Start in test mode**".
7. Click "**Enable**".

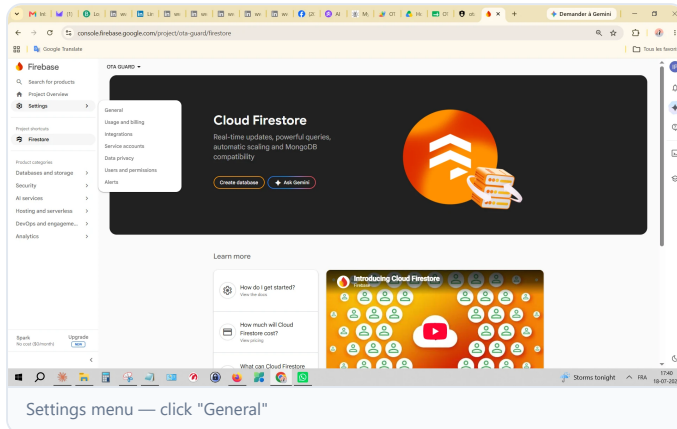
Note: test mode leaves the database open for 30 days — enough for setup and initial testing. Final security rules (restricting access to the owning hotel only) will be added before any real client deployment.

3

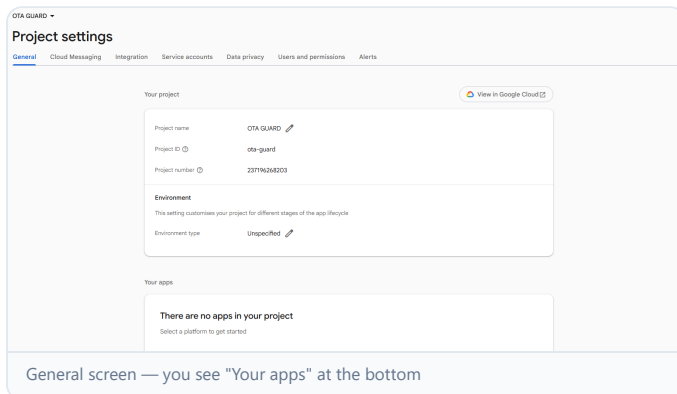
Get the client configuration (5 values)

These 5 values fill the 5 fields in OTA Guard's **"Mobile App"** tab (PC) to generate the pairing QR code. They are not secret.

1. Click the gear icon ⚙️ (top left) → in the dropdown menu, click **"General"**.

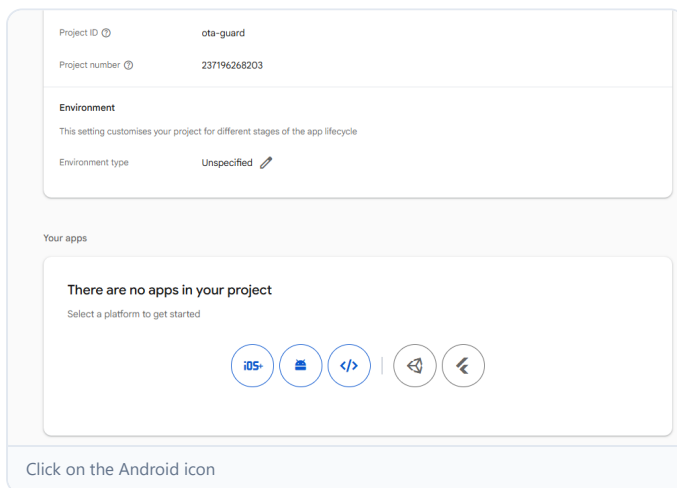


Settings menu — click "General"



General screen — you see "Your apps" at the bottom

2. Scroll down to **"Your apps"** → click the Android icon (the robot 🤖).



Click on the Android icon

3. Enter exactly the package name: `com.example.ota_guard_mobile` (it's the default name of the generated Flutter app).

Add Firebase to your Android app



You can also register an Android app with an AI coding agent

[Try it](#)

1 Register app

Android package name ⓘ

App nickname(optional) ⓘ

Register app

2 Download and then add config file

3 Add Firebase SDK

4 Next steps

Form — fill in the package name and nickname, then click "Register app"

4. Nickname: `OTA Guard Mobile` → click **"Register app"**.

Google services Gradle plug-in.

☒ Kotlin DSL (build.gradle.kts) ☐ Groovy (build.gradle)

Add the plug-in 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 "4.5.0" apply false  
}
```

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

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

```
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:34.16.0"))  
  
    // 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.

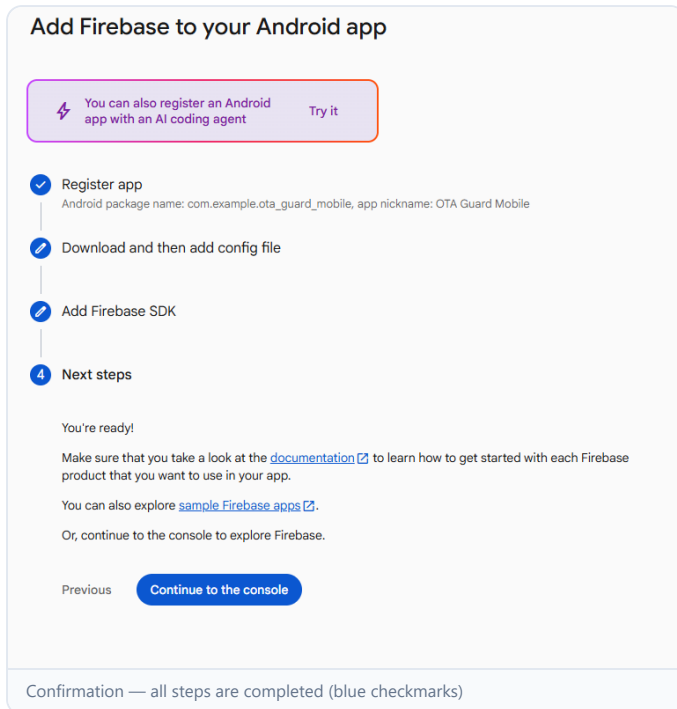
[Learn more](#)

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

Step 5 — Kotlin DSL or Groovy choice

5. You see a proposal to download `google-services.json` and configure Gradle (Kotlin DSL or Groovy) → **you can ignore this section completely**. Simply click **"Next"** to continue.

6. Click **"Next"** again. You see a confirmation screen **"Add Firebase to your Android app"** with all steps completed (blue checkmarks).



7. Click **"Continue to the console"** (blue button) to return to the project settings.

8. Back in Project settings, you will see the 5 values:

- apiKey
- appId
- messagingSenderId ("Sender ID")
- projectId
- storageBucket

What to do next:

1. Copy these 5 values
2. Open **OTA Guard** (the PC application)
3. Go to the **"Mobile App"** tab
4. Fill the 5 fields with these values
5. Click **"Save"**

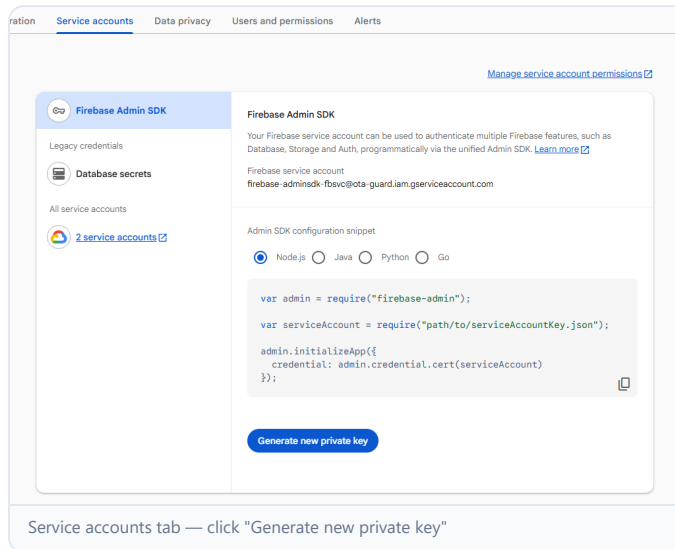
4

Generate the service account key (file for the PC)

4

Generate the service account key (PC file)

1. Project settings → "Service accounts" tab.



2. Click "Generate new private key" → confirm.

3. A `.json` file downloads automatically.

4. In OTA Guard, "📱 Mobile App" tab → "Load JSON file" → select this file.

5. Click "Test connection" → status should switch to "Connected".

Important: this `.json` file is a secret credential (equivalent to a password). Never send it by plain email, never share it publicly. OTA Guard stores it encrypted once loaded.

WHO PERFORMS THESE 4 STEPS?

ACCOUNT REQUIRED

- These steps must be done with the **hotel's own Google account** (not the vendor's).
- This is what keeps the hotel's data isolated and Firebase hosting free of charge.

TWO WAYS TO PROCEED

- **Option A:** the vendor clicks through on the client's behalf (video call / screen share), once.
- **Option B:** the client follows this guide alone, step by step.