



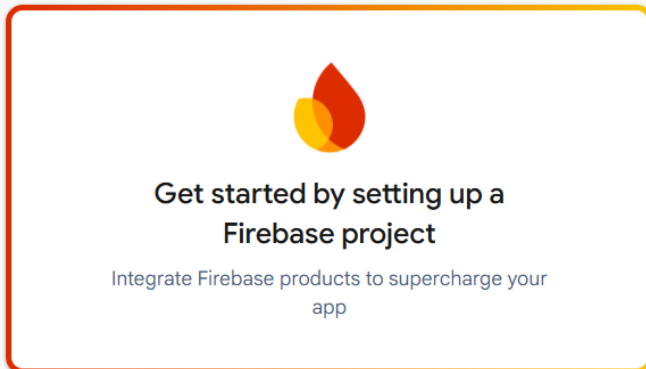
Ce guide explique comment créer un projet **Firebase** gratuit — nécessaire pour activer l'application mobile compagnon d'OTA Guard (alertes en temps réel, contrôle des prix depuis le téléphone). Google ne demande **aucune carte bancaire** pour ces étapes. Prévoir environ 10 minutes.

1

Créer le projet Firebase

1. Rendez-vous sur <https://console.firebase.google.com/>, connecté avec le compte Google de l'hôtel.

Get started



Écran d'accueil — cliquez sur la carte « Get started by setting up a Firebase project »

2. Cliquez sur la carte « **Get started by setting up a Firebase project** ».
3. Saisissez un nom de projet — Firebase génère automatiquement en dessous un identifiant technique à partir de ce nom (modifiable si besoin).
4. Cochez la case « **I accept the Firebase terms** » (obligatoire).
5. Laissez désactivé le bouton « **Join the Google Developer Programme** » (non nécessaire).
6. Cliquez sur « **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

Nommage du projet et validation des conditions Firebase

7. Sur l'écran « **AI assistance for your Firebase project** » (Gemini in Firebase), laissez le bouton « **Enable Gemini in Firebase** » désactivé — non nécessaire pour OTA Guard, même si Google le marque « Recommended » — puis cliquez sur « **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

Proposition Gemini in Firebase — à laisser désactivée

8. Sur l'écran « **Google Analytics for your Firebase project** », laissez le bouton « **Enable Google Analytics for this project** » désactivé (non nécessaire ici).

9. Cliquez sur « **Create project** » et patientez ~30 secondes.

×

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

Proposition Google Analytics — à laisser désactivée

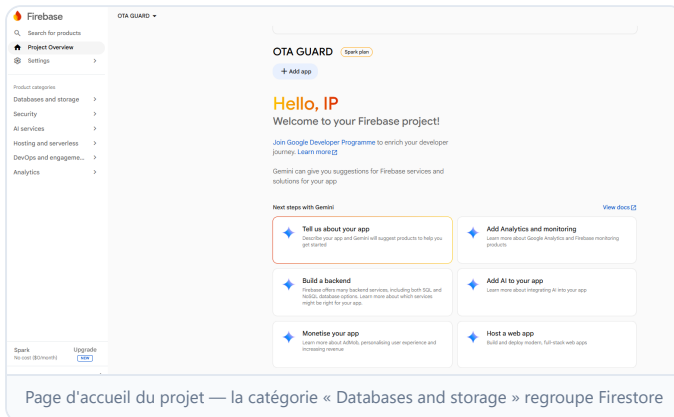
EXEMPLE

Nom de projet suggéré : **ota-guard-jangala** (préfixe fixe + nom de l'hôtel, en minuscules, sans espace)

2

Activer la base de données Firestore

1. Une fois le projet créé, vous arrivez sur la page d'accueil du projet (« Hello... Welcome to your Firebase project »).
2. Dans le menu de gauche, sous « **Product categories** », cliquez sur « **Databases and storage** » pour dérouler la catégorie (elle est repliée par défaut — Firestore n'apparaît pas directement dans le menu).



3. Cliquez sur « **Firestore** » (sous la section « **NoSQL** ») qui apparaît dans la liste déroulée.
4. Cliquez sur « **Créer une base de données** ».
5. Choisissez une région proche de l'hôtel (ex. `asia-south1` pour l'Inde, `eur3` pour l'Europe).
6. Sélectionnez « **Démarrer en mode test** ».
7. Cliquez sur « **Activer** ».

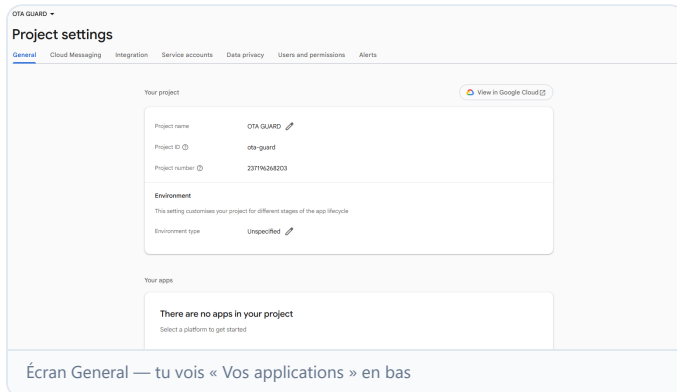
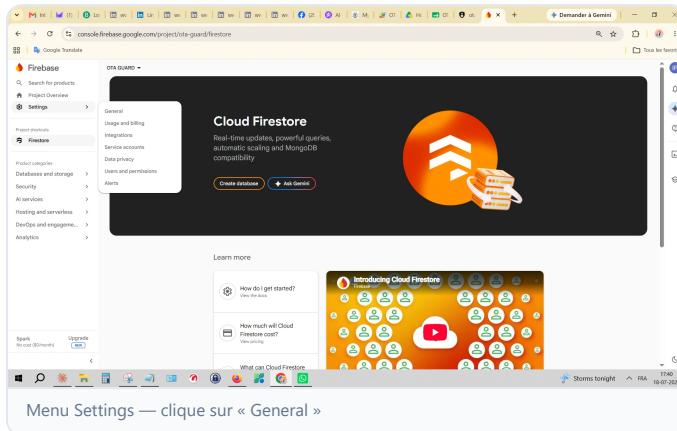
À savoir : le mode test ouvre l'accès à la base pendant 30 jours — suffisant pour la mise en route et les premiers tests. Des règles de sécurité définitives (restreignant l'accès au seul hôtel propriétaire) seront ajoutées avant tout déploiement client final.

3

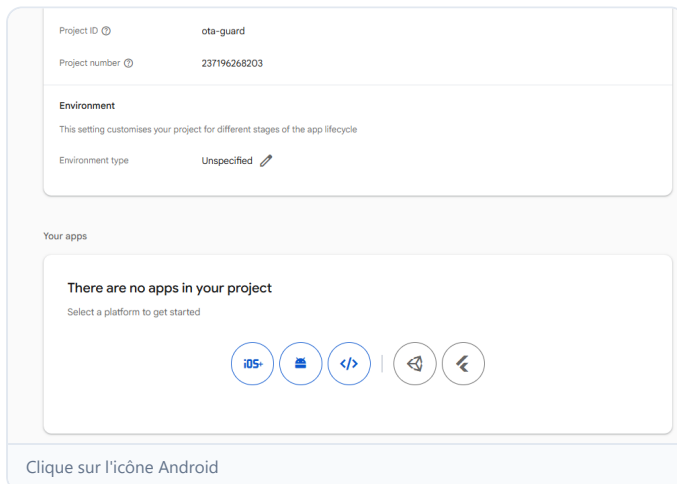
Récupérer la configuration client (5 valeurs)

Ces 5 valeurs remplissent les 5 champs de l'onglet « **App mobile** » d'OTA Guard (PC) pour générer le QR code d'appairage. Elles ne sont pas secrètes.

1. Clique sur l'engrenage ⚙️ (en haut à gauche) → dans le menu déroulant, clique sur « **General** ».



2. Descends jusqu'à « **Vos applications** » → clique sur l'icône Android (le petit robot 🤖).



3. Tape exactement le nom du package : `com.example.ota_guard_mobile` (c'est le nom par défaut de l'app Flutter générée).

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

Formulaire — remplis le package name et le nickname, puis clique « Enregistrer »

4. Surnom : `OTA Guard Mobile` → clique « **Enregistrer l'application** ».

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.

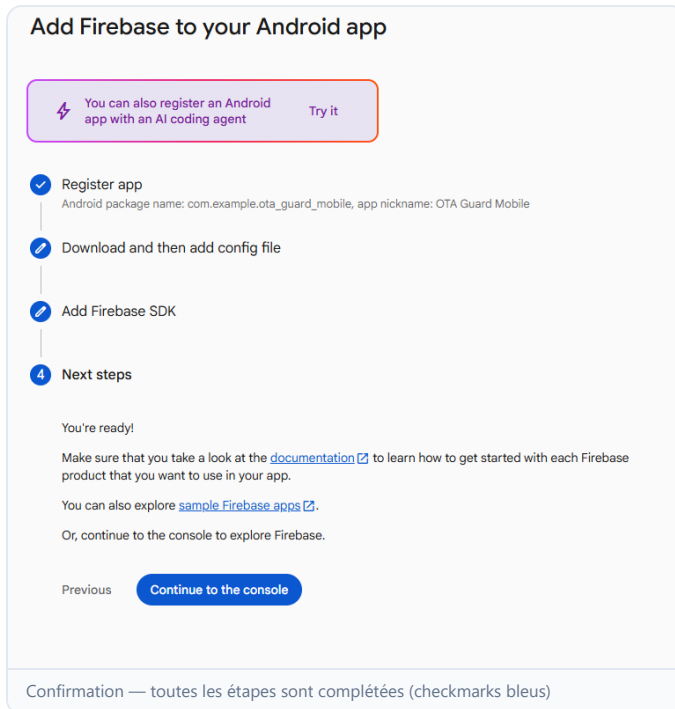
[Learn more](#)

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

Étape 5 — choix Kotlin DSL ou Groovy

5. Tu vois une proposition de télécharger `google-services.json` et de configurer Gradle (Kotlin DSL ou Groovy) → **tu peux ignorer cette section complètement**. Clique simplement « **Suivant** » pour continuer.

6. Clique « **Suivant** » encore une fois. Tu vois un écran de confirmation « **Add Firebase to your Android app** » avec les étapes complétées (checkmarks bleus).



7. Clique « **Continue to the console** » (bouton bleu) pour retourner aux paramètres du projet.

8. De retour dans Paramètres du projet, tu verras les 5 valeurs :

- o apiKey
- o appId
- o messagingSenderId (« ID de l'expéditeur »)
- o projectId
- o storageBucket

Ce qu'il faut faire ensuite :

1. Copie ces 5 valeurs
2. Ouvre **OTA Guard** (l'application PC)
3. Va dans l'onglet « **App mobile** »
4. Remplis les 5 champs avec ces valeurs
5. Clique « **Sauvegarder** »

4

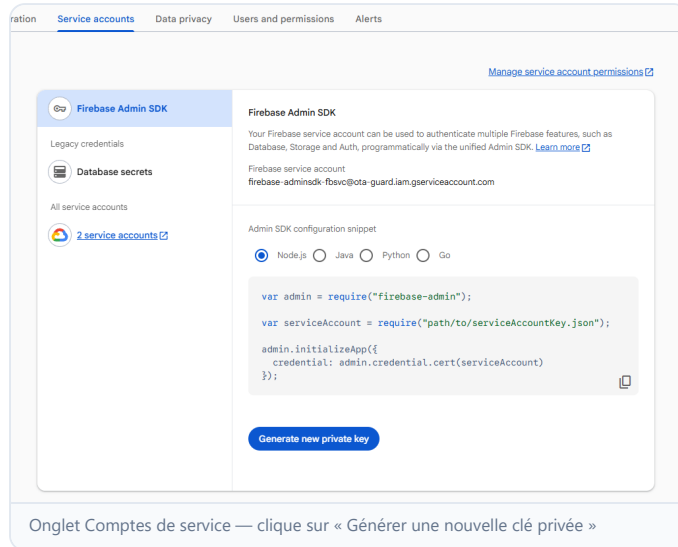
Générer la clé de service (fichier pour le PC)

4

Générer la clé de service (fichier pour le PC)

C'est le fichier secret que charge le bouton «  **Charger le fichier JSON** » de l'onglet App mobile.

1. Toujours dans Paramètres du projet → onglet « **Comptes de service** ».



2. Clique « **Générer une nouvelle clé privée** » → confirme.

3. Un fichier `.json` se télécharge → dans OTA Guard, onglet «  **App mobile** » → clique «  **Charger le fichier JSON** » → sélectionne ce fichier.

4. Le statut doit passer à  **Compte de service chargé**.

5. Clique «  **Tester la connexion** » → doit afficher « **Connexion établie** ».

Important : ce fichier `.json` est un identifiant secret (équivalent d'un mot de passe). Ne jamais l'envoyer par email en clair, ne jamais le partager publiquement. OTA Guard le stocke chiffré une fois chargé.

QUI RÉALISE CES ÉTAPES ?

COMPTE REQUIS

- Ces étapes doivent se faire avec le **compte Google de l'hôtel** (pas celui du vendeur).
- C'est ce qui garantit que les données de l'hôtel restent isolées et que l'hébergement Firebase reste gratuit.

DEUX FAÇONS DE PROCÉDER

- **Option A :** le vendeur clique à la place du client (visio / partage d'écran), une seule fois.
- **Option B :** le client suit ce guide seul, étape par étape.