Build an Android app for your shop

The website works independently of this optional app. Start with the Android source archive in Admin → APK Builder. Use your own Firebase project, Android application ID, signing identity and shop domain. Once distributed, keep the application ID and signing key for every subsequent release.

Configure the source

1. Choose a unique application ID such as com.yourbusiness.shop. Register that exact ID as an Android app in your own Firebase project. Download the Android client file google-services.json.

2. In the extracted Android project, run:

```
python3 tools/configure-customer.py com.yourbusiness.shop \
https://shop.example.com /path/to/google-services.json
python3 tools/check-project.py
```

3. Set the Android app name, icon and version in your shop's Android branding settings for the Admin build flows. If building outside Admin, set the manifest label, resources and version in your own source. Test on your device before publishing.

The Android app file and the Firebase server service account JSON are different files. Keep the service account JSON in private server storage. Use the documented Firebase permissions needed for Cloud Messaging and Remote Config for your own project. On the Ubuntu shop server:

```
cd /var/www/gemma-creations
sudo bash install/configure-firebase.sh /path/to/service-account.json
```

Review Android settings in Admin, publish Remote Config and test push registration and delivery on a device. Keep push disabled until you confirm the correct account, app and domain receive notifications.

Choose a build method

- GitHub Actions. Store the configured source and the included workflows in a new private repository. Use the signing initializer for a new app or import your existing identity. Set the repository's release-signing passphrase and upload URL/token from the shop's APK Builder settings. Save your repository and dispatch credentials in Admin. Keep keys and tokens out of source control. Check workflow logs and verify the first release on a device. If you also build on Ubuntu, use the same permanent signing identity there.

- Ubuntu server builder. Configure the installed copy under /var/www/gemma-creations/android with the same tool and identifiers before scheduling a build. Set up SDK dependencies and signing through install/install-android-build-prereqs.sh and install/setup-android-server-builder.sh. The server builder validates the imported project's application ID, shop origin, Firebase project and signing fingerprint before accepting a build. Check the signed APK on a device before publication.

- Android Studio or local Gradle. Use Java 17, Gradle 8.9 and Android SDK Platform 35 with Build Tools 35.0.0. Configure release signing on your own workstation. Check the signed package ID, signing fingerprint and version before uploading it through your chosen distribution method.

Firebase Android setup documentation