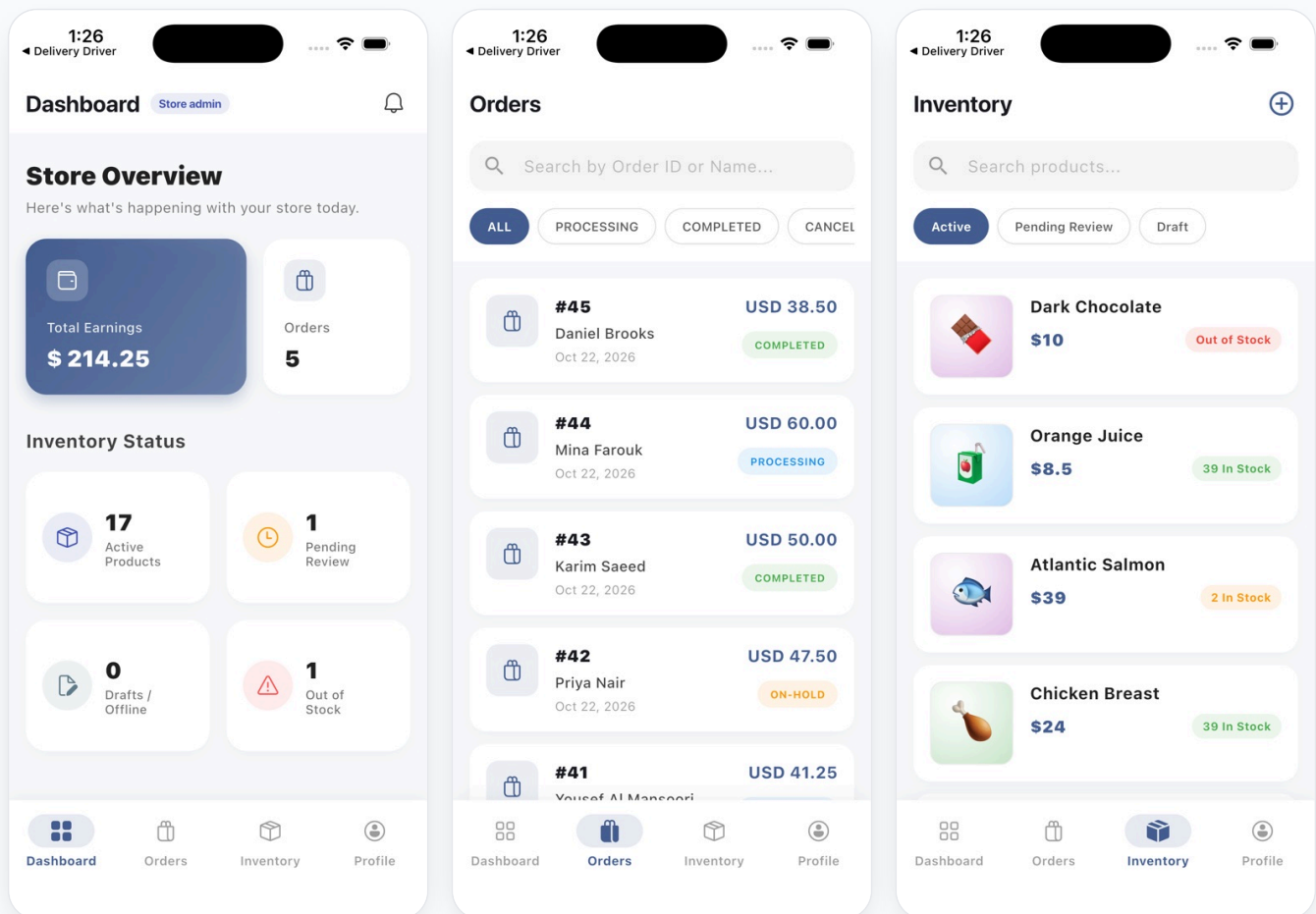


Multivendor Seller App

A step-by-step guide for everyone, no coding needed: install the plugin, understand who can log in, set up push notifications, use the app as a seller or as the store admin, and build your own Android app online. Developers will also find the source-code and REST API chapters.

1. Start here

In plain words: This product puts your **marketplace in your pocket**. Your sellers, and you as the store admin, log into a phone app with their normal WordPress account to see orders, add products, edit their store and get a notification when a new order arrives.



The seller app: dashboard, orders and products.

What you bought: three parts that work together

Part	What it is	Who needs it
1. The plugin multivendor-app-for-woocommerce-plugin.zip	Goes on your WordPress store. It gives the app its data, checks every login, sends push notifications and SMS, and lets you build the app.	Everybody
2. The online App Builder (a tab inside the plugin)	Makes your own branded Android app (your name, icon and colours) on our servers. You fill in a short form and press one button.	Most people
3. The app source code flutter-app-source.zip	The complete code of the app (Flutter, for Android and iPhone), for developers who want to change it or build it themselves.	Developers

Tip: The store admin can use the app too. Log in with your normal admin account (for example `admin`) and you see the **whole** store. The quickest way to try everything is to start with your admin login, as shown in this guide.

2. Who can log in, and what they see

In plain words: The app asks WordPress who you are. What you see depends on your role.

Role	Can log in?	Sees
Administrator	Yes	All orders and all products of the store. A small “Store admin” badge shows on the dashboard.
Shop manager	Yes	Same as administrator.
Seller / vendor (Dokan <i>seller</i> , WCFM <i>wcfm_vendor</i> , MultiVendorX <i>dc_vendor</i> or <i>vendor</i>)	Yes	Only their own products and the orders that contain their products.
Customer, subscriber, anyone else	No	“You do not have vendor access.”

The plugin detects your marketplace plugin by itself:

Marketplace plugin	What the app uses
Dokan	Dokan orders, earnings and store profile
WCFM Marketplace	WCFM orders, commissions, store profile and policies
MultiVendorX	MultiVendorX orders, earnings and vendor profile
None installed	Admin and shop manager login only

Important: Sellers only see orders when one of the three marketplace plugins is active, because the plugin needs it to know which order belongs to which seller. Administrators do not need one.

3. What you need before you start

You need	Why
A WordPress site with WooCommerce 8.0 or newer , PHP 7.4+	The plugin reads and updates your WooCommerce orders and products.
For sellers: Dokan , WCFM Marketplace or MultiVendorX	Tells the plugin which products and orders belong to which seller.
HTTPS (a padlock in the browser) and “pretty” permalinks	The phone talks to your store through https://yourstore.com/wp-json/ .
A free Firebase project (Google account)	Needed for push notifications. Chapter 7 shows every click.
Optional: an SMS provider account (Twilio, Unifonic, 2Factor or OTPless)	Texts for new orders.
Your purchase code	Needed by the online App Builder (chapter 9).

4. Install the plugin

- 1 Unzip the package you downloaded. Inside are `multivendor-app-for-woocommerce-plugin.zip` and `flutter-app-source.zip`. Do **not** unzip these two.
- 2 In WordPress open **Plugins → Add New → Upload Plugin**, choose `multivendor-app-for-woocommerce-plugin.zip`, press **Install Now**, then **Activate**.

- 3 Open **Settings → Seller App**. If you see the page with six tabs (App Config, Branding, SMS & WhatsApp, Push Notifications, Test Notifications, App Builder), the plugin is ready.
- 4 Check **Settings → Permalinks**: anything except "Plain" is fine. Press *Save Changes* once if the app cannot reach your store.

Important: If you used the older "MultiVendor Seller App for WooCommerce" 1.0.0 or the separate "Taqnix App Builder" plugin, read chapter 14 first: this plugin replaces both.

5. The settings, tab by tab

In plain words: Open **Settings → Seller App**. Change what you need and press **Save Settings**. The app reads these rules when it starts.

App Config

Seller App Configuration

App Config

Branding

SMS & WhatsApp

Push Notifications

Test Notifications

App Builder

App Config & Updates

Maintenance Mode

☐ Lock the mobile app temporarily. Vendors will see a maintenance message.

Maintenance Message

Shown on the lock screen while Maintenance Mode is on.

New Products by Sellers

Publish immediately

▼

What happens to a product a seller adds in the app. Store admins always publish at once.

App Update Link

Where the "Update Now" button of the forced-update screen goes (your Google Play or App Store page).

Force App Update (Min Version)

Lowest allowed version (e.g., 1.0.5). Older app users will be forced to update.

Privacy Policy URL

Required for Apple/Google approval (e.g., https://yourstore.com/privacy).

Save Settings

App Config tab.

Setting	What it does
Maintenance Mode + Message	Locks the app with a “Back soon” screen showing your message. Use it while you work on the store.
New Products by Sellers	<i>Publish immediately</i> , or <i>Pending review</i> so you approve every product a seller adds. Store admins always publish at once.
App Update Link	Where the “Update now” button goes: your Google Play or App Store page.
Force App Update (Min Version)	The lowest app version allowed, for example <code>1.2.0</code> . Older apps show an “Update required” screen and cannot be used.
Privacy Policy URL	Required by Google Play and the App Store.

Branding

Seller App Configuration

App Config

Branding

SMS & WhatsApp

Push Notifications

Test Notifications

App Builder

Branding & Support

App Primary Color

Hex color for buttons and headers.

Support Email

support@grocer.example

Email for vendors to contact you.

Support WhatsApp

+971455501188

Include country code (e.g., +919876543210).

Branding tab.

Primary colour, support email and support WhatsApp number. The app can read them from `/app-config` .

SMS & WhatsApp

Seller App Configuration

App Config

Branding

SMS & WhatsApp

Push Notifications

Test Notifications

App Builder

Vendor SMS & WhatsApp

Enable Automated Alerts ☒ Automatically send text/WhatsApp to vendors when they receive a new order.

SMS Provider

Twilio (Global SMS & WhatsApp) ▼

Twilio / Unifonic API Key

ACxxxxxxxxxxxxxxxxxxxxxxxx

Generic API Key / Client ID.

Twilio API Secret

Secret password or Auth Token.

2Factor API Key

Enter your 2Factor API Key.

OTPless Client ID

Your OTPless Client ID.

OTPless Client Secret

Your OTPless Client Secret.

Sender Name (Header)

GRO CER

Your approved Sender ID (e.g., TAQNIX).

Provider Template ID

Template ID/Name if using restricted templates for OTPless or 2Factor.

**New Order Message
Template**

SMS & WhatsApp tab.

Switch on **Enable Automated Alerts**, choose a provider, paste the key (and secret), the sender name and edit the message. You can use `{order_id}` , `{total}` , `{store_name}` , `{customer_name}` , `{payment_method}` and `{order_date}` . The seller gets this text when an order that contains their products is placed. The seller's phone number comes from the billing phone on their WordPress profile.

Push Notifications

Seller App Configuration

App Config

Branding

SMS & WhatsApp

Push Notifications

Test Notifications

App Builder

Firestore Push Notifications

Firestore Service Account
JSON

```
{
  "type": "service_account",
  "project_id": "your-app-name"
}
```

Firestore connects your website to the vendor's app for push notifications.

Save Settings

Push Notifications tab: the Firestore service account goes here.

Test Notifications

Seller App Configuration

App Config

Branding

SMS & WhatsApp

Push Notifications

Test Notifications

App Builder

Push Notification & SMS Sandbox

Send a direct test to a user's mobile device to ensure your API keys and Firestore JSON are working properly.

Search User

Search by name or email. Note: The user must have logged into the Flutter app at least once to receive push notifications.

Test Message

Hello! This is a test notification from TaqniX.

Send Actions



Send Test Push



Send Test SMS

Save Settings

Send a test push or SMS to any user.

Search a user, type a message and press **Send Test Push** or **Send Test SMS**. The answer from Firebase or your SMS provider is shown on the page, which makes problems easy to find. The user must have logged into the app once to receive a push.

Tip: Secrets such as the SMS secret are never shown again after you save them. Leave the box empty to keep the saved one.

6. Set up push notifications (Firebase)

In plain words: Push notifications are the “ding” on the seller’s phone when a new order arrives. Google provides them for free through **Firebase**. You set this up once. It needs two files: one for the *app*, one for your *store*.

A. Create the Firebase project

- 1 Go to console.firebase.google.com and sign in with a Google account.
- 2 Press **Add project**, type a name (for example “Acme Sellers”) and finish the wizard. You can turn Google Analytics off.

B. The file for the app: `google-services.json` (Android)

- 1 In your project press the **Android** icon (“Add app”).
- 2 **Android package name:** type the unique name of your app, for example `com.acme.sellers`. Write it down: use **exactly** the same name in the App Builder (chapter 9).
- 3 Press **Register app**, then **Download google-services.json**. Keep it for chapter 9 or 10. Skip the remaining Firebase screens.

C. The file for your store: a service account key

- 1 In Firebase press the gear icon → **Project settings** → tab **Service accounts**.
- 2 Press **Generate new private key** and confirm. A `.json` file downloads.
- 3 Open it in a text editor, select **everything** and copy.
- 4 In WordPress open **Settings** → **Seller App** → **Push Notifications**, paste into the box and press **Save Settings**.

Important: The service-account file is a password for your Firebase project. Do not email it or put it in a public place.

D. iPhone only

Also add an **iOS** app in Firebase (use your bundle ID), download [GoogleService-Info.plist](#) and upload an Apple **APNs key** under *Project settings* → *Cloud Messaging*. Details are in chapter 10.

Test it

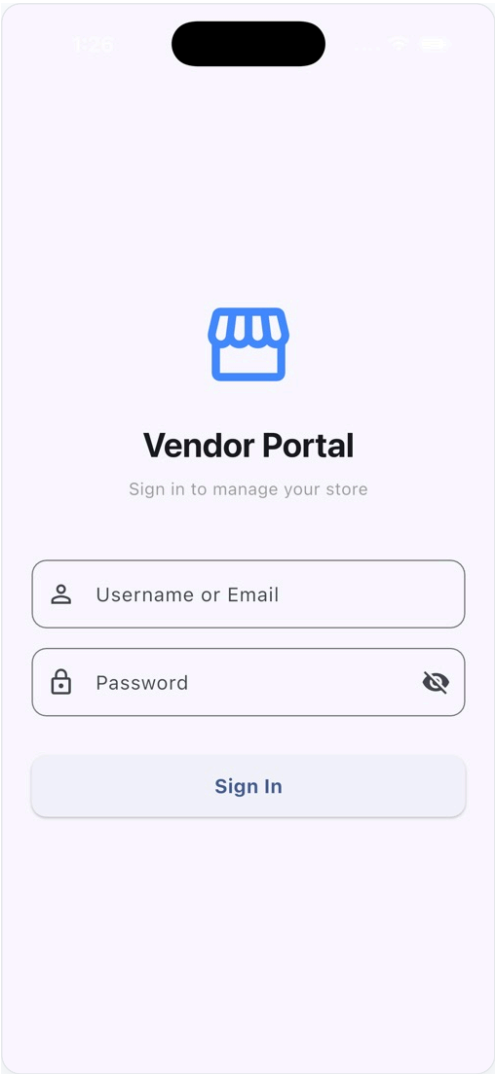
Log into the app on a phone, then open **Test Notifications**, pick that user and press **Send Test Push**. The phone should ring within seconds.

7. Using the app

In plain words: The app has four tabs along the bottom: **Dashboard**, **Orders**, **Inventory** (your products) and **Profile**.

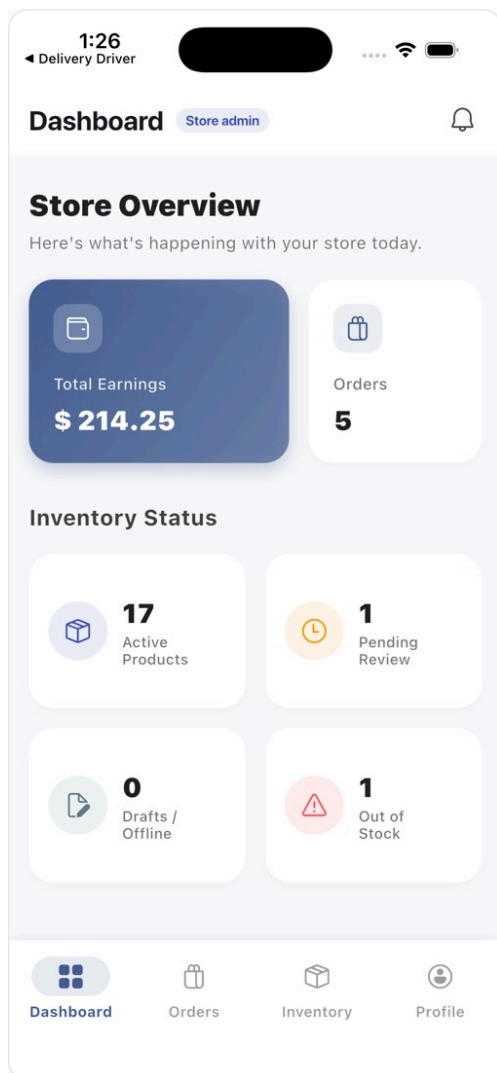
Log in

- 1 Open the app and type your WordPress **username** and **password**. Store admins use their admin login.
- 2 Allow **notifications** when the phone asks.
- 3 You stay logged in on this phone until you press **Sign Out** (or 90 days pass, or the role is removed). Logging in on a second phone does not log out the first.



The login screen.

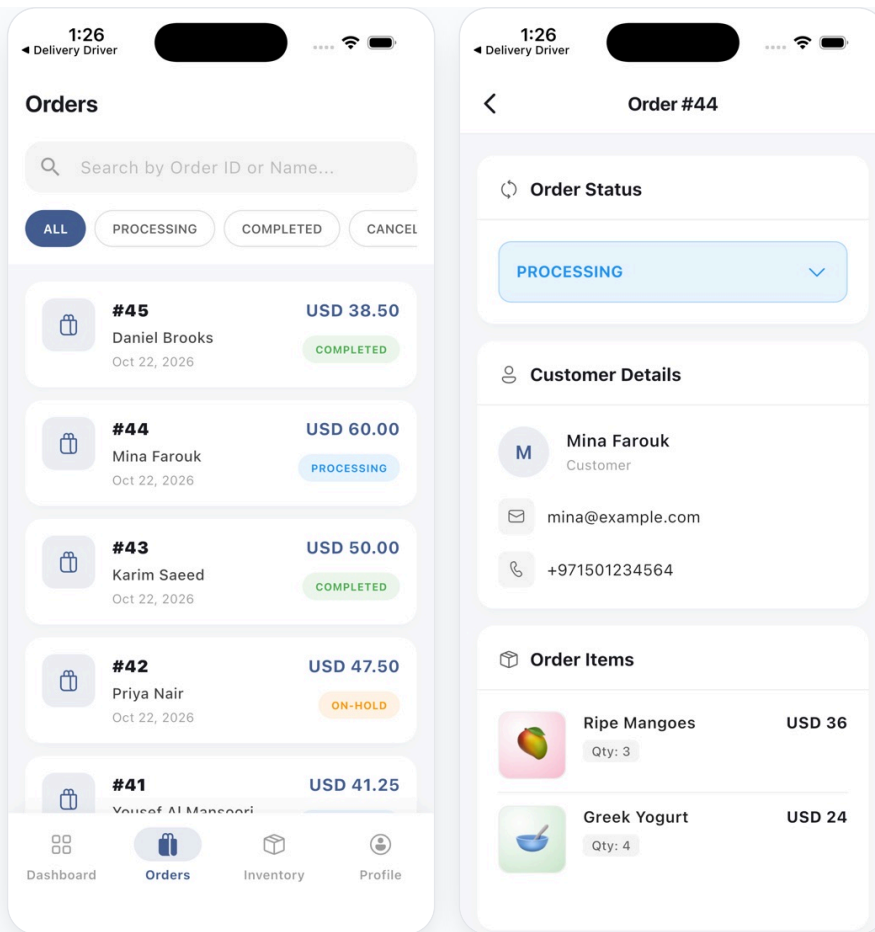
Dashboard



Dashboard: earnings, orders, products and stock alerts.

Total earnings (store admins see the total sales of the whole store), number of orders, active, pending and offline products, and products that are out of stock. Pull down to refresh. The bell shows alerts such as out-of-stock or low-stock products and products waiting for approval. A "Store admin" badge shows when you are logged in as admin.

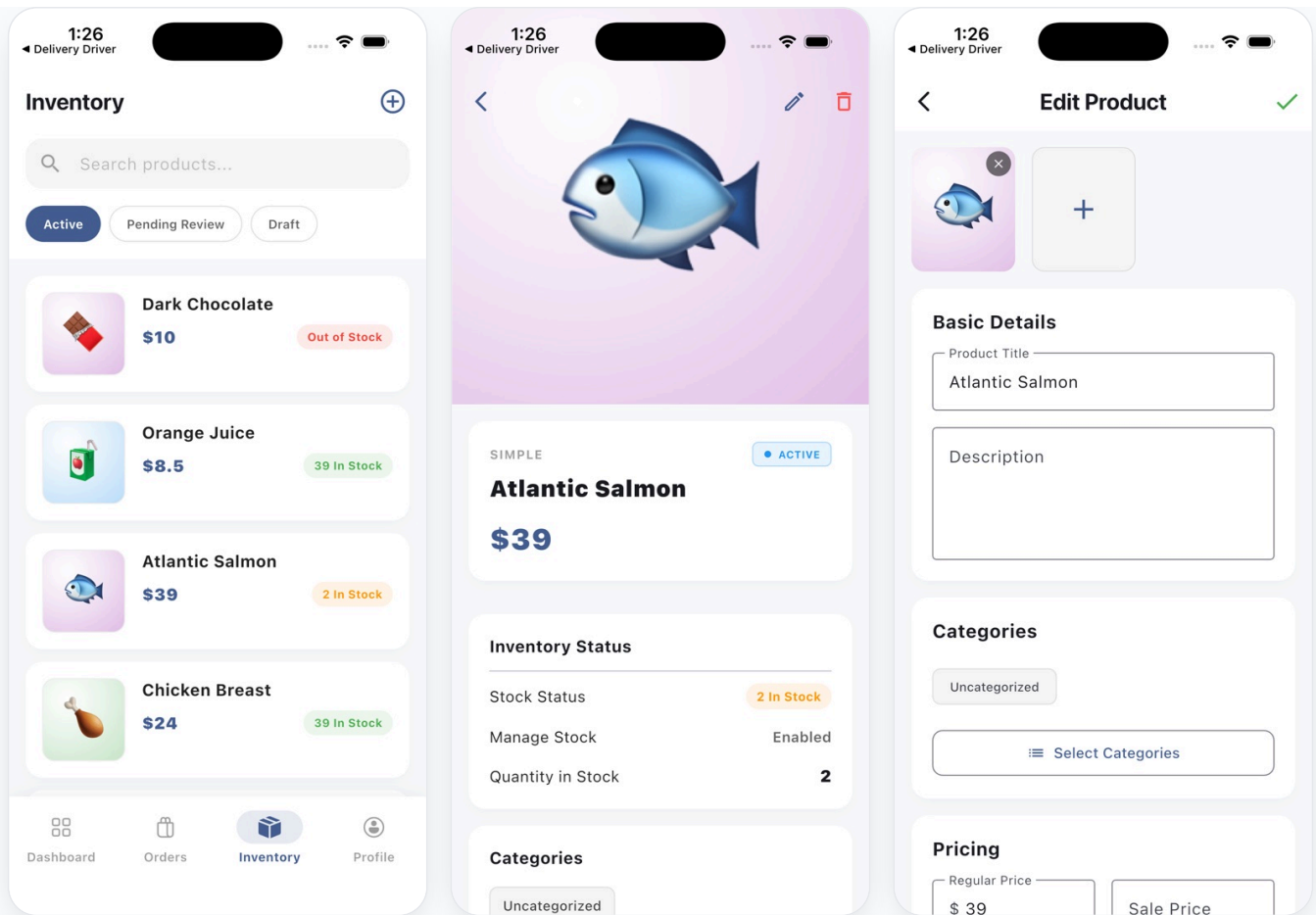
Orders



Orders: search, filter by status, open an order and change its status.

- Search by order number (admins also by customer name or email) and filter by status.
- Open an order to see the customer, payment method, products with photos and the total.
- Change the status with the **Order Status** box at the top (it is a drop-down list), for example from *Processing* to *Completed*. Sellers can use *Processing*, *On hold*, *Completed* and *Cancelled*. Only store admins can use other statuses such as *Refunded*.

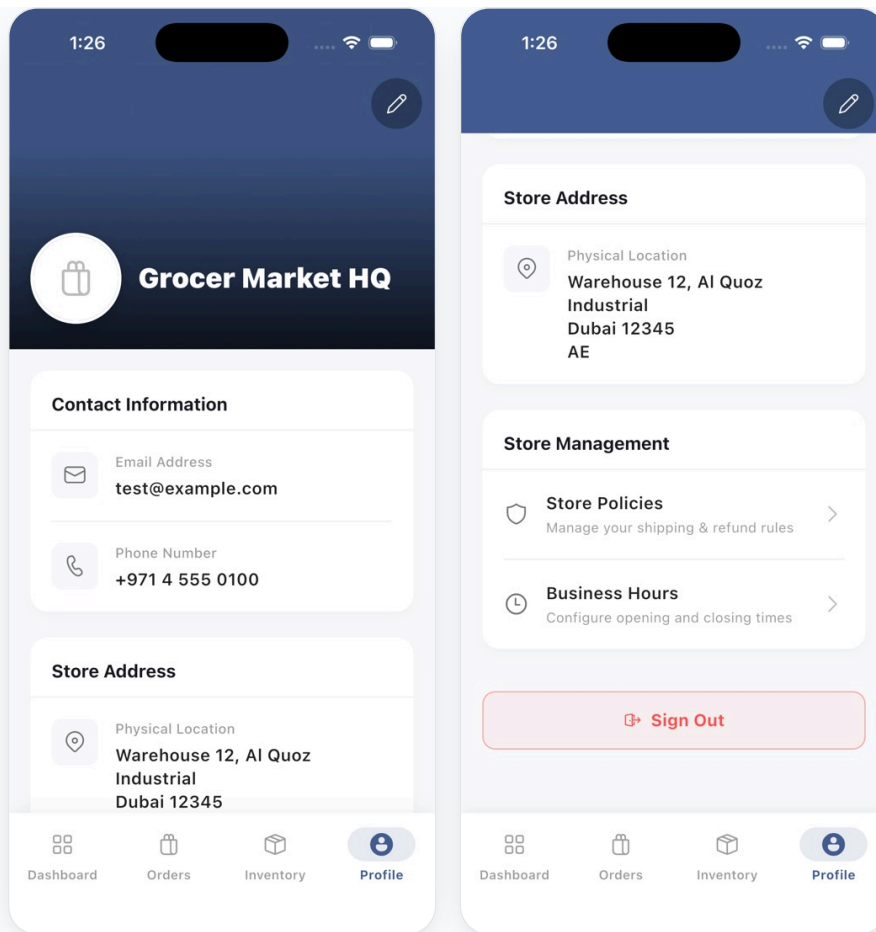
Inventory (products)



Inventory: list, product details and editing.

- Search, filter by status, and use **Out of stock** from the dashboard.
- Tap the round **+** at the top right to add a product: name, prices, sale price, description, categories, stock quantity or stock status, a main photo and gallery photos from the camera or gallery.
- Open a product and use the pencil to edit it or the bin to delete it. Deleted products go to the trash.
- Sellers only see and change their **own** products; store admins see all.

Profile and store settings



Profile: contact details, address, store policies and business hours.

Edit the store name, phone, address, logo and banner, **business hours** and the shipping, refund and cancellation **policies**.

8. Order alerts for sellers and admins

When a customer places an order, the plugin finds which sellers own products in it and sends each of them a **push notification** ("New Order Received! Order #123 total is ...") and, if switched on, an **SMS**. Store administrators are notified of every order. Each order is announced only once.

A seller's phone must be registered: the app does this by itself after login and every time the app opens. A phone belongs to one account: if somebody else logs in on the same phone, the previous account stops receiving alerts from it. Signing out also stops alerts for that phone.

Tip: Dokan and similar plugins split one customer order into one order per seller. Each seller is alerted about their part.

9. Build your own branded app online (no coding)

In plain words: The **App Builder** tab makes an Android file (APK) with *your* app name, icon and colours, pointing at *your* store. You fill in a form; a few minutes later you get an email with the download link. You do not need to install anything.

Seller App Configuration

[App Config](#)[Branding](#)[SMS & WhatsApp](#)[Push Notifications](#)[Test Notifications](#)[App Builder](#)

Connected. Your app settings are loaded below.

Step 1 · Connect your purchase code

Enter your website address and the purchase code you received with your order. Your code is stored on this server and is never shown again.

Website address

Purchase code

[Connect & load my app](#)

● Connected

Step 2 · Name your app

App name

The name under the icon on the seller's phone, for example "Acme Sellers".

Package name

A unique ID for Google Play: lowercase letters and dots, like com.yourbrand.sellers. You cannot change it after publishing.

Your email

We email you the download link when your app is ready.

Phone

Step 3 · Add your branding

App icon 1024×1024

[Choose File](#) No file chosen

Logo

[Choose File](#) No file chosen

Shown in the middle of the screen while the app opens.

Splash image

[Choose File](#) No file chosen

Settings → Seller App → App Builder.

Before you start

- Your **purchase code**.

- Your app **icon** (square PNG, 1024×1024), a **logo** and a **splash image**.
- The `google-services.json` file from chapter 6, made for the package name you will type below.

Step by step

- 1 Open **Settings** → **Seller App** → **App Builder**.
- 2 **Step 1.** Check your website address (https, no slash at the end), paste your purchase code and press **Connect & load my app**. The green “Connected” sign appears. Your code stays on your server and is not shown again.
- 3 **Step 2.** Type the **App name** (shown under the icon), the **Package name** (the same one you gave Firebase, for example `com.acme.sellers`) and your email. The download link is sent there.
- 4 **Step 3.** Choose the icon, logo and splash images (they upload as soon as you pick them), choose the splash colour and upload `google-services.json` .
- 5 **Step 4.** Choose **APK** (install straight on a phone) or **App Bundle** (for Google Play). Press **Save details**, then **Start Android build**.
- 6 Wait for the email (usually a few minutes; builds are queued). Open the link on the phone to download and install the APK.

Important: The package name in the builder and in Firebase must be **identical**. If they differ, the build stops with a message that tells you which names it found.

Good to know

- The builder makes **Android** apps. For iPhone use the source code (chapter 10).
- Every purchase code can build one app. To change the name, icon or screens later, save the form again and press Build again.
- What is sent to the build server: your site address, purchase code, app name, package name, your email and phone, splash colour, your three images and the Firebase file. No store passwords or WooCommerce keys are needed.
- Only administrators can use this tab; every request is protected by a security token.

10. For developers: build the app from source

In plain words: Skip this chapter if you use the online builder. The full source of the app is in `flutter-app-source.zip` : Flutter (Dart), Riverpod, go_router, Dio, Firebase Messaging, Secure Storage, Image Picker.

Requirements

- Flutter 3.41 or newer (`flutter doctor` green), Android Studio for Android, Xcode and CocoaPods for iPhone.

1. Open the project and point it to your store

```
unzip flutter-app-source.zip -d seller_app && cd seller_app
flutter pub get

# run on a connected phone or emulator
flutter run --dart-define=API_DOMAIN=https://yourmarketplace.com
```

Or change the default permanently in `lib/core/constants/app_config.dart` :

```
class AppConfig {
  static const String siteUrl = String.fromEnvironment(
    'API_DOMAIN',
    defaultValue: 'https://yourmarketplace.com', // <- your store, no slash at the end
  );
  static const String apiBaseUrl = '$siteUrl/wp-json/multivendor-seller/v1';
}
```

2. Add your Firebase files

- Android: `android/app/google-services.json` (the Gradle plugin switches on automatically when the file exists).
- iPhone: `ios/Runner/GoogleService-Info.plist` ; open `ios/Runner.xcworkspace` in Xcode and drag it into the Runner group if it shows in red.

3. Name, package and icons

```
# unique app id (must match Firebase)
dart run change_app_package_name:main com.acme.sellers

# replace assets/images/icon.png (1024x1024) and assets/images/splash.png, then:
dart run flutter_launcher_icons
dart run flutter_native_splash:create
```

The app name shown on the phone is in `android/app/src/main/AndroidManifest.xml` (`android:label`) and `ios/Runner/Info.plist` (`CFBundleDisplayName`).

4. Sign and build (Android)

```
# create a signing key once and keep it safe
keytool -genkey -v -keystore ~/acme-sellers.jks -keyalg RSA -keysize 2048 -validity 10000 -alias upload

# android/key.properties
storeFile=/Users/you/acme-sellers.jks
storePassword=...
keyAlias=upload
keyPassword=...
```

In `android/app/build.gradle.kts` read these values in a `signingConfigs { create("release") { ... } }` block and use it in `buildTypes.release` instead of the debug key (the file has a comment where). Then:

```
flutter build apk --release --dart-define=API_DOMAIN=https://yourmarketplace.com # APK to test
flutter build appbundle --release --dart-define=API_DOMAIN=https://yourmarketplace.com # AAB for Google P
```

5. iPhone

```
cd ios && pod install --repo-update && cd ..
flutter build ipa --release --dart-define=API_DOMAIN=https://yourmarketplace.com
```

In Xcode choose your **Team**, set a unique **Bundle Identifier** (the same as in Firebase) and keep the *Push Notifications* capability. Upload an APNs key in Firebase.

Where things are

Folder	What is inside
<code>lib/core/api/api_client.dart</code>	Network client: adds the login token to every call and returns to the login screen when the server ends the login.
<code>lib/features/auth</code>	Login, splash screen (maintenance and forced-update checks), session.
<code>lib/features/dashboard</code>	Dashboard and the bottom navigation.
<code>lib/features/orders</code> , <code>products</code>	Order and product lists, details, add and edit.
<code>lib/features/profile</code>	Store profile, policies, business hours.
<code>lib/features/notifications</code>	In-app alert list.
<code>lib/core/services/fcm_service.dart</code>	Registers the phone for push messages.
<code>default_files/</code>	Templates used by the online builder. Ignore.

11. For developers: the REST API

In plain words: The app and the plugin talk through a small REST API. You can use it for your own app or integrations.

Base address: `https://yourstore.com/wp-json/multivendor-seller/v1` . Everything except `/login` and `/app-config` needs the token from login, sent as `Authorization: Bearer ...` (or in the header `X-Seller-Token`). Tokens last 90 days; each phone has its own (up to 5 per user).

Method and path	What it does
POST /login	username , password → token , user_id , display_name , is_admin . 429 after 5 wrong passwords.
POST /logout	Ends this token; optional fcm_token is forgotten too.
GET /app-config	Public: maintenance mode and message, minimum version, update link, currency, colour, support and legal links.
GET /dashboard-stats	Earnings, orders, product counts, out-of-stock.
GET /seller-orders	page , per_page (max 50), status , search .
PUT /seller-orders/{id}	status : processing, on-hold, completed, cancelled (admins: any real status).
GET /seller-products	page , per_page , status , search , stock_status .
POST /seller-products , PUT /seller-products/{id}	title , regular_price , sale_price , description , short_description , manage_stock , stock_quantity , stock_status , categories , image_id , gallery_image_ids .
DELETE /seller-products/{id}	Trash (add force=1 to delete forever).
GET /store-categories	Product categories.
POST /upload-media	Multipart file : JPG, PNG, GIF or WebP → attachment_id , url .
GET/PUT /seller-profile	Store name, phone, address, logo/banner, policies, business hours.
GET/PUT /seller-settings	Policies and vacation mode (for your own apps; the included app edits policies in the profile).
GET /seller-notifications	In-app alert list.
POST /seller-notifications/token	fcm_token of this phone.

Try it

```
# log in (admin works too)
curl -s -X POST https://yourstore.com/wp-json/multivendor-seller/v1/login \
  -H 'Content-Type: application/json' \
  -d '{"username":"admin","password":"your-password"}'
# -> {"success":true,"token":"MXx8...", "user_id":1, "is_admin":true,...}

# list orders
curl -s 'https://yourstore.com/wp-json/multivendor-seller/v1/seller-orders?status=processing' \
  -H 'Authorization: Bearer MXx8...'

# complete an order
curl -s -X PUT https://yourstore.com/wp-json/multivendor-seller/v1/seller-orders/123 \
  -H 'Authorization: Bearer MXx8...' -H 'Content-Type: application/json' -d '{"status":"completed"}'
```

Let another role use the app

```
// functions.php or a small plugin
add_filter( 'mvs_app_roles', function ( $roles ) {
    $roles[] = 'my_custom_vendor_role';
    return $roles;
} );
```

12. Security and privacy

- **Every request needs a login token** (except login and the public app settings). A seller can only read and change their own products, orders, profile and photos; the server decides, not the app.
- Tokens are random, kept as a one-way hash, last 90 days, are per phone, and stop working at once when the user is logged out, deleted or loses the role.
- Login attempts are limited (5 wrong passwords, then 15 minutes).
- Uploads must be real JPG, PNG, GIF or WebP images; prices, statuses and stock values are checked.
- Sellers cannot refund or trash orders from the app.
- The App Builder tab and every admin action need an administrator *and* a security token; nothing is available to visitors.
- The SMS and Firebase secrets stay on your server. The SMS secret is not printed back into the settings page.
- Tell your sellers in your privacy policy what the app handles (orders, customer names and phone numbers, product photos).

13. Troubleshooting

Problem	What to do
"Cannot reach the server"	Check the store address in the app (<code>API_DOMAIN</code>), HTTPS, and open <code>https://yourstore.com/wp-json/multivendor-seller/v1/app-config</code> in a browser: you should see text starting with <code>{"success":true}</code> . If not, re-save <i>Settings</i> → <i>Permalinks</i> .
"You do not have vendor access"	The user's role must be administrator, shop manager or a seller role of your marketplace plugin (not Customer or Subscriber).
"Too many failed attempts"	Wait 15 minutes.
The app suddenly shows the login screen	The login ended (signed out elsewhere, role changed, 90 days). Log in again.
A seller sees no orders	Dokan, WCFM or MultiVendorX must be active and the order must contain that seller's products. Try the same login as admin to compare.
The admin dashboard shows 0 orders	Orders in the trash, cancelled, refunded or failed are not counted. Open WooCommerce → Orders to compare.
No push notification	Use <i>Test Notifications</i> and read the answer. Check the service-account file (chapter 6C), that the user logged in on that phone, and that notifications are allowed on the phone.
Product photos do not upload	Only JPG, PNG, GIF and WebP are accepted. Check your server's upload size limit.
"Update required" or "Back soon" screen	Settings → Seller App → App Config: lower the minimum version or switch off Maintenance Mode.
The App Builder says "Not allowed"	Your admin session expired: reload the page.
The App Builder says the package name does not match	Make the package name in the builder identical to the one in Firebase, or download a new <code>google-services.json</code> .
Cache plugins (LiteSpeed, WP Rocket, Cloudflare)	Exclude <code>/wp-json/multivendor-seller/</code> from caching.

14. Updating, and moving from the older plugins

Updating

Upload the new plugin zip with **Replace current with uploaded**; settings stay. Build a new app version (chapter 9 or 10) and raise *Force App Update* when you want everybody to update.

If you used “MultiVendor Seller App for WooCommerce” 1.0.0 and “Taqnix App Builder”

- This plugin **replaces both**. Deactivate and delete “Taqnix App Builder”; its builder is now the *App Builder* tab here (enter your purchase code once).
- The old plugin file had a stray apostrophe in its name. Delete the old plugin folder completely, then upload this one.
- **Everybody has to log in again once**. Login tokens are stored differently now (per phone, with an expiry), so old sessions end.
- Settings are kept. The new options (maintenance message, new-product review, update link) start with safe defaults.

Removing

Deactivate and delete the plugin. Sellers, products and orders stay. The plugin’s settings are removed.

15. Support

Installation help and bug fixes are included for 6 months, with lifetime updates. Contact us through the marketplace item page where you bought the package. Please include your store address, which marketplace plugin you use, what you did, and what you saw (a screenshot helps).

Changelog

1.1.0 – Initial marketplace release

- Flutter seller app for Android and iOS: login for sellers and store admins, dashboard, orders (search, filter, status), inventory (add, edit, delete, photos), profile with policies and business hours, in-app alerts, push notifications
- WordPress plugin 1.1.0: secure token REST API for Dokan, WCFM and MultiVendorX, admin and shop manager login, push and SMS order alerts, maintenance mode and forced update, review of new seller products
- App Builder tab (merged from the Taqnix App Builder plugin): build your Android APK or App Bundle online
- Security: one token per phone with expiry and logout, login throttling, sellers limited to their own data, image-only uploads, validated statuses, prices and stock
- Complete Flutter source for Android and iOS with the Android release permissions and the optional Firebase Gradle plugin