

Hop In Express Command OS

Technical Architecture & API Reference

Generated: 19/1/2026

1. System Architecture Overview

The Hop In Express "Command OS" uses a Hybrid Architecture designed for high-reliability retail environments. It combines a local Node.js automation server with a cloud-based React frontend.

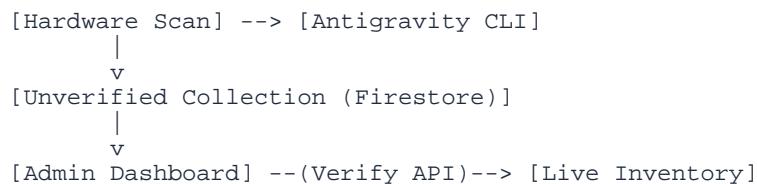
Core Components:

- **Frontend (Vite/React):** The visual interface running in the browser. Handles user interactions, real-time data visualization, and direct Firestore subscriptions.
- **Local Automation Server (Express):** Runs on port 3001. Handles hardware interactions (scanners), file system operations (image uploads), and "privileged" verification tasks.
- **Data Layer (Firebase):** Firestore serves as the central source of truth for Inventory, Staff, and Sales. Firebase Auth handles identity.

2. Key Workflows & Data Privacy

Workflow A: Inventory Ingestion (The "Gatekeeper" Pattern)

Data Privacy: High. Unverified items stay in a staging status. Only authorized admins can promote items to "LIVE".



Workflow B: Staff Onboarding

1. User registers via Auth View (Firebase Auth).
2. User enters "Pending" state in Firestore.
3. Admin verifies credentials via the "Verify Staff" API.
4. Staff gains access to the Terminal.

Workflow C: Sales & Transactions

All sales are atomic. When a sale occurs, a "Batch Write" is sent to Firestore that simultaneously:

- Creates a Transaction Record.
- Decrement stock from the specific Inventory Item.
- Updates the Ledger (if applicable).

3. API Reference (Local Express Server)

Base URL: <http://localhost:3001>

Automation & Hardware

POST /shelf-scan/start
 > Triggers the Antigravity hardware shelf scanner for UK Grocery mode.

Administrative Verification

```
GET /admin/unverified-products  
    > Fetches items waiting in the staging area.  
POST /admin/verify-product/:id  
    > Promotes an item to LIVE status working inventory.  
POST /admin/verify-staff/:id  
    > Activates a pending staff member account.
```

Asset Management

```
POST /product/upload-image/:id  
    > Uploads product image to local storage and links it in DB.
```

4. Database Schema (Firestore)

```
shops/{id}/inventory  
    Main product catalog. Fields: sku, stock, price_gbp, vatRate.  
shops/{id}/transactions  
    Immutable sales history. Fields: items[], total, timestamp.  
shops/{id}/staff  
    Staff profiles and roles. Fields: name, pin, role, status.  
shops/{id}/attendance  
    Clock-in/out logs for payroll.  
shops/{id}/ledger  
    Double-entry bookkeeping records for expenses and sales.
```

CONFIDENTIAL - HOP IN EXPRESS INTERNAL DOCS