# Serapod QR Code for Cursor V2

**Purpose.** This Markdown gives Cursor AI concrete, production‑oriented work orders that align 1:1 with our **UI Blueprint v2** and **Functional Build Pack v2**. It focuses on code‑generation targets: endpoints, payloads, validations, batch math, PO rendering, notifications, and verify behavior.

## 0) Tech Targets & Envs

* **Runtime:** Supabase **Edge Functions** (Deno/TypeScript).
* **DB:** Supabase Postgres with **RLS**; sequences for codes; Storage for PDFs/SVG/CSV/XLSX/images.
* **Web Admin:** Next.js (App Router).
* **Mobile Scanners:** React Native (Expo), offline queue (SQLite), idempotent ingest.
* **PDF:** pdf-lib (edge‑only). CSV/XLSX: xlsx. SVG labels via server render.

**Environment variables**

SUPABASE\_URL  
SUPABASE\_ANON\_KEY  
SERVICE\_ROLE\_KEY (CI only)  
BASE\_SHORT\_URL=https://q.serapod2u.com  
WHATSAPP\_TOKEN (optional)  
SMTP\_HOST/SMTP\_USER/SMTP\_PASS (optional)

**API conventions**

* Base path: /api/v1/...
* Success: { ok:true, data, request\_id }
* Error: { ok:false, error:{ code, message, field\_errors?, hint? }, request\_id }
* Clients send X-PG-ID (Product Group). Edge functions call select set\_config('app.pg\_id', :pg\_id, true) before DB work.

## 1) Shared Utilities (generate once)

* **ULID generator** (masters/units).
* **Idempotency key** helper: hash(pg\_id + code + device\_id + device\_seq).
* **CSV/XLSX** helpers (labels & manifests).
* **PDF** helpers using pdf-lib (A4 metrics, table/grid primitives).
* **Signed URL** helper for Storage.
* **Response envelope** & **error mappers** (HTTP→domain codes).
* **HMAC‑SHA256 PIN hasher** + per‑batch salt (never store plaintext).

## 2) Work Order — QR Settings & Batch Creation

**Endpoint:** POST /api/v1/batches

**Preconditions**

* Order is **L2‑approved** and **payment verified** (HQ→Manufacturer flow).
* Product is **locked** after first batch.

**Payload**

{  
 "pg\_id": "uuid",  
 "order\_id": "uuid",  
 "product\_id": "uuid",  
 "total\_units": 10000,  
 "production\_date": "2025-01-01",  
 "mfg\_facility\_code": "MY",  
 "master\_id\_mode": "both",  
 "label\_badge\_id": null,  
 "label\_badge\_text\_override": "FREE DEVICE",  
 "label\_badge\_apply\_to": "unit"  
}

**Logic** (config‑driven)

UPM = qr\_settings.units\_per\_master  
B1000 = qr\_settings.buffer\_per\_1000  
BufferUnits = floor(total\_units/1000) \* B1000  
TotalUniqueQRs = total\_units + BufferUnits  
MastersCount = ceil(TotalUniqueQRs / UPM)

* Generate **ULIDs** for masters/units; set seq\_no / seq\_in\_master.
* Persist in a **single transaction**; set batches.masters\_count.
* Render **CSV/XLSX** (verify links, mapping, EPC placeholders) and **PDF/SVG** labels; upload to Storage; return **signed URLs**.

**Response (201)**

{  
 "ok": true,  
 "data": {  
 "batch": {  
 "id": "...",  
 "batch\_no": 23001567,  
 "masters\_count": 55,  
 "settings": { "units\_per\_master": 200, "buffer\_per\_1000": 100 },  
 "master\_id\_mode": "both"  
 },  
 "export\_urls": {"csv":"...","xlsx":"...","pdf":"...","svg\_zip":"..."}  
 },  
 "request\_id": "req\_789"  
}

**Validation**

* Enums: master\_id\_mode in {qr\_only, rfid\_only, both}; label\_badge\_apply\_to in {unit, master, both}.
* Dates valid; total\_units > 0; product/order belong to caller’s **pg\_id**.
* Reject if order status < payment\_verified.

## 3) Work Order — PO Render & Send

**Endpoints**

* POST /api/v1/orders/:id/po/render → create/overwrite purchase\_order\_documents with **PDF** (and optional UBL XML).
* POST /api/v1/orders/:id/po/send → choose channels (email/WhatsApp), check **HQ Settings** caps, log sends, set po\_sent\_at.

**PDF Layout (A4, matches sample)**

* Header: company + **PURCHASE ORDER** title, **PO#**, **Date**.
* Party blocks: **Supplier** (manufacturer) vs **Ship To**.
* Items table: **No, Description, Qty, Unit Price, Amount**.
* Totals: **Subtotal, Tax, Total (MYR)**; signature lines (**Issued by / Accepted by**).
* Fonts Helvetica 10pt; grid lines; thousands separators; 2 decimals.

**Decision:** MVP uses `` (Edge‑only). Switch to HTML→PDF later if pixel‑perfect branding is required.

## 4) Work Order — Product Status & Catalog

**Endpoints**

* PATCH /api/v1/products/:id/status *(HQ)*: { pg\_id, status: 'active'|'archived'|'blocked' }
* GET /api/v1/catalog/products?pg\_id=&q=&brand\_id=&category\_id= *(Distributor/Shop)* → returns \*\*only \*\*`` with: id, code, name, brand, category, image\_url, flavour, price\_cents, country\_of\_origin, weight\_g.

**Rules**

* Non‑HQ cannot see archived/blocked products.
* Catalog may show a **badge** (e.g., FREE DEVICE) if an upcoming order/batch carries a label badge.

## 5) Work Order — Distributor→HQ Orders

**Flow** pending\_review → approved → dpo\_sent → hq\_invoice\_sent → payment\_acknowledged → completed

**Endpoints**

* POST /api/v1/distributor-orders
* POST /api/v1/distributor-orders/:id/approve *(HQ)* / .../reject
* POST /api/v1/distributor-orders/:id/send-dpo *(Distributor)* → creates PO doc (direction distributor\_to\_hq)
* POST /api/v1/distributor-orders/:id/send-invoice *(HQ)*
* POST /api/v1/distributor-orders/:id/ack-payment *(Distributor)*
* GET /api/v1/distributor-orders/:id/tracking → steps for **food‑delivery style** tracker

**Notifications**

* shop\_new\_batch\_coming (optional, HQ toggled) with template:  
  *“New batch is coming. Order & contact your distributor now. {{free\_gift\_note}}”*
* Variables: {{product\_name}}, {{brand}}, {{category}}, {{badge\_text}}, {{free\_gift\_note}} (auto‑empty when no badge).

## 6) Work Order — Verify (Profile‑Derived)

* vape ⇒ verify ends at **Shop/Kedai**; consumer scans go to **health‑warning** landing page.
* nonvape ⇒ **consumer path** allowed only when **ActiveCampaign(unit)**.

**Next.js route**: /verify/[code] resolves ULID/EPC, loads group profile, renders the appropriate panel. Include a **mock mode** for zero‑data prototypes.

## 7) Validation & Error Codes

**Global**

* Require X-PG-ID and valid JWT.
* Enforce RLS by forwarding Authorization header to Supabase client.
* Map HTTP→domain codes: VALIDATION\_ERROR, UNAUTHORIZED, FORBIDDEN, NOT\_FOUND, CONFLICT, PRECONDITION\_FAILED, UNPROCESSABLE\_ENTITY, RATE\_LIMITED, IDEMPOTENT\_REPLAY, INTERNAL\_ERROR.

**Endpoint‑specific** (examples)

* /products: brand/category must match, **image required** on create, Option‑B for vape/regulated categories.
* /batches: status ≥ payment\_verified; enums valid; cohesive pg\_id.
* Distributor orders enforce legal transitions; wrong role ⇒ FORBIDDEN.

## 8) Tests

* **Unit:** buffer math, masters\_count, currency formatting, envelope helpers.
* **Integration:** RLS scope via set\_config, product visibility by status, order approvals preconditions.
* **E2E:** POST /batches creates rows + assets; /orders/:id/po/render stores PDF and returns signed URL.

## 9) Stretch (deferred)

* UBL 2.1 for POs/Invoices; Email‑to‑Inbox ingestion; OCR; ERP webhooks; AR UI after verify; ML anomaly detection.