# Serapods — Dev Bootstrap Seeds (standalone)

**Purpose.** One-click(ish) starter data and runbook so engineers/AI can boot a dev environment and immediately exercise **MDM, Orders, Batches, PO render**. Idempotent: safe to re-run.

## A) Prereqs

* Supabase project created; Database **extensions** enabled: pgcrypto, uuid-ossp (default OK in Supabase).
* CLI/tools: node >=20, pnpm, git, supabase CLI, psql.
* Auth providers (later): turn on Google/Facebook only when you need consumer flows.

**Env**

SUPABASE\_URL=  
SUPABASE\_SERVICE\_ROLE\_KEY=  
DATABASE\_URL=postgresql://postgres:<password>@<host>:<port>/postgres

## B) Where to put seeds in the repo

repo/  
 supabase/  
 migrations/ # your DDL migrations  
 seed.sql # paste Section C here  
 apps/admin/ # Next.js (UI)  
 apps/api/ # Edge Functions

Run order:

supabase db push # apply schema/migrations  
psql "$DATABASE\_URL" -f supabase/seed.sql # load dev data

## C) seed.sql (copy‑paste)

Aligns with **Functional Build Pack v2** DDL. If any tables are renamed in migrations, adjust names below.

-- === Product Groups (profile ⇒ derived verify) ===  
insert into product\_groups (code,name,workflow\_profile,verify\_behavior)  
values ('VAPE','Vape','vape','auth\_till\_kedai'),  
 ('CAMP','Camping','nonvape','auth\_till\_consumer')  
on conflict (code) do nothing;  
  
-- === QR Settings (config-driven batch math) ===  
insert into qr\_settings (product\_group\_id, units\_per\_master, buffer\_per\_1000)  
select id, 200, 100 from product\_groups where code='VAPE'  
on conflict (product\_group\_id) do update set units\_per\_master=excluded.units\_per\_master, buffer\_per\_1000=excluded.buffer\_per\_1000;  
  
insert into qr\_settings (product\_group\_id, units\_per\_master, buffer\_per\_1000)  
select id, 100, 50 from product\_groups where code='CAMP'  
on conflict (product\_group\_id) do update set units\_per\_master=excluded.units\_per\_master, buffer\_per\_1000=excluded.buffer\_per\_1000;  
  
-- === Manufacturers (HQ→Manufacturer flow) ===  
insert into manufacturers (product\_group\_id, code, name, email, phone, address, country)  
select id, 'MFR-SERAPOD', 'Serapod Factory', 'ops@factory.example', '+60-12-0000000', 'Somewhere, MY', 'MY'  
from product\_groups where code='VAPE'  
on conflict (code) do nothing;  
  
-- === Categories & Brands (VAPE) ===  
with pg as (  
 select id from product\_groups where code='VAPE'  
)  
insert into categories (product\_group\_id, code, name, requires\_regulatory)  
select pg.id, 'CAT-CLOSEPOD', 'ClosePod', true from pg  
on conflict (product\_group\_id, name) do nothing;  
  
with ctx as (  
 select pg.id as pg\_id, c.id as cat\_id  
 from product\_groups pg  
 join categories c on c.product\_group\_id=pg.id and c.code='CAT-CLOSEPOD'  
 where pg.code='VAPE'  
)  
insert into brands (product\_group\_id, category\_id, code, name)  
select ctx.pg\_id, ctx.cat\_id, 'BRD-SERAX', 'SeraX' from ctx  
on conflict (product\_group\_id, category\_id, name) do nothing;  
  
-- === Example Product (tangible attrs + image placeholder) ===  
with ctx as (  
 select pg.id as pg\_id, c.id as cat\_id, b.id as brand\_id, m.id as mfr\_id  
 from product\_groups pg  
 join categories c on c.product\_group\_id=pg.id and c.code='CAT-CLOSEPOD'  
 join brands b on b.product\_group\_id=pg.id and b.code='BRD-SERAX'  
 left join manufacturers m on m.product\_group\_id=pg.id and m.code='MFR-SERAPOD'  
 where pg.code='VAPE'  
)  
insert into products (  
 product\_group\_id, category\_id, brand\_id, manufacturer\_id,  
 code, name, image\_url,  
 short\_description, long\_description,  
 flavour, price\_cents, country\_of\_origin, weight\_g,  
 nicotine\_free, nicotine\_strength\_mg\_per\_ml, volume\_ml, child\_resistant\_packaging  
)  
select ctx.pg\_id, ctx.cat\_id, ctx.brand\_id, ctx.mfr\_id,  
 'PRD-000001', 'SeraX Strawberry 30ml', 'https://placehold.co/512x512.png?text=Product',  
 'Strawberry vape e-liquid', 'Smooth strawberry profile',  
 'Strawberry', 1999, 'MY', 120,  
 false, 12, 30, true  
from ctx  
on conflict (code) do nothing;  
  
-- === Optional: Camping group basic taxonomy ===  
with pg as (select id from product\_groups where code='CAMP')  
insert into categories (product\_group\_id, code, name)  
select pg.id, 'CAT-GADGET', 'Gadgets' from pg  
on conflict (product\_group\_id, name) do nothing;  
  
with ctx as (  
 select pg.id as pg\_id, c.id as cat\_id from product\_groups pg  
 join categories c on c.product\_group\_id=pg.id and c.code='CAT-GADGET'  
 where pg.code='CAMP'  
)  
insert into brands (product\_group\_id, category\_id, code, name)  
select ctx.pg\_id, ctx.cat\_id, 'BRD-SERAOUT', 'SeraOutdoor' from ctx  
on conflict (product\_group\_id, category\_id, name) do nothing;  
  
-- === Sample HQ & Manufacturer roles (replace UUIDs) ===  
-- select \* from auth.users; -- get UUIDs, then:  
-- insert into user\_group\_roles (user\_id, product\_group\_id, role)  
-- select '<HQ\_USER\_UUID>', id, 'hq' from product\_groups where code in ('VAPE','CAMP');  
-- insert into user\_group\_roles (user\_id, product\_group\_id, role)  
-- select '<MFR\_USER\_UUID>', id, 'manufacturer' from product\_groups where code='VAPE';  
  
-- === Optional: Sample order lines (manual code for demo) ===  
-- Create an order awaiting approvals  
with ctx as (  
 select g.id as pg\_id, m.id as mfr\_id, p.id as prod\_id  
 from product\_groups g  
 join manufacturers m on m.product\_group\_id=g.id and m.code='MFR-SERAPOD'  
 join products p on p.product\_group\_id=g.id and p.code='PRD-000001'  
 where g.code='VAPE'  
)  
insert into orders (product\_group\_id, code, manufacturer\_id, notes)  
select ctx.pg\_id, 'PO-DEMO-000001', ctx.mfr\_id, 'Demo order for batch testing' from ctx  
on conflict (code) do nothing;  
  
-- Mark approvals (simulate L1/L2)  
with o as (select id from orders where code='PO-DEMO-000001')  
insert into order\_approvals (order\_id, approval\_type, approver\_id, approver\_role)  
select o.id, 'hq\_l1', gen\_random\_uuid(), 'hq' from o  
on conflict do nothing;  
with o as (select id from orders where code='PO-DEMO-000001')  
insert into order\_approvals (order\_id, approval\_type, approver\_id, approver\_role)  
select o.id, 'hq\_l2', gen\_random\_uuid(), 'hq' from o  
on conflict do nothing;  
  
-- Advance order status to approved  
update orders set status='approved', approved\_by\_id\_1=gen\_random\_uuid(), approved\_at\_1=now(),  
 approved\_by\_id\_2=gen\_random\_uuid(), approved\_at\_2=now()  
where code='PO-DEMO-000001' and status='pending\_approval';

Next: use the **/api/v1/orders/:id/po/render** and **/api/v1/orders/:id/po/send** endpoints, then create a batch with **POST /api/v1/batches** (Manufacturer role) using this order.

## D) Runbook (step‑by‑step)

1. **Apply schema & seeds**

supabase db push  
psql "$DATABASE\_URL" -f supabase/seed.sql

1. **Assign roles** (replace UUIDs) — run the commented user\_group\_roles inserts for one HQ user and one Manufacturer user.
2. **Log in as HQ** → create additional Categories/Brands/Products and upload real images.
3. **Create PO PDF** for demo order

curl -X POST \  
 -H "Authorization: Bearer <HQ\_JWT>" \  
 -H "X-PG-ID: <VAPE\_GROUP\_UUID>" \  
 https://<project>.functions.supabase.co/api/v1/orders/<ORDER\_UUID>/po/render

1. **Send PO** (email/WhatsApp controlled in HQ Settings)

curl -X POST \  
 -H "Authorization: Bearer <HQ\_JWT>" \  
 -H "X-PG-ID: <VAPE\_GROUP\_UUID>" \  
 https://<project>.functions.supabase.co/api/v1/orders/<ORDER\_UUID>/po/send

1. **As Manufacturer**, after payment verified → create **Batch**

curl -X POST \  
 -H "Authorization: Bearer <MFR\_JWT>" \  
 -H "X-PG-ID: <VAPE\_GROUP\_UUID>" \  
 -H 'Content-Type: application/json' \  
 -d '{  
 "pg\_id":"<VAPE\_GROUP\_UUID>",  
 "order\_id":"<ORDER\_UUID>",  
 "product\_id":"<PRODUCT\_UUID>",  
 "total\_units": 10000,  
 "production\_date":"2025-01-01",  
 "mfg\_facility\_code":"MY",  
 "master\_id\_mode":"both",  
 "label\_badge\_text\_override":"FREE DEVICE",  
 "label\_badge\_apply\_to":"unit"  
 }' \  
 https://<project>.functions.supabase.co/api/v1/batches

1. **Download exports** from the batch response (csv/xlsx/pdf/svg\_zip URLs) and test verify pages.

## E) Notes & Safety

* **Never** run seeds in production. These are demo only.
* All amounts are in **MYR** using integer **cents**.
* Replace placeholder image URLs with your Storage signed URLs during real testing.
* If you alter DDL names in migrations, update this seed accordingly.