

ClearKeep — Ops Runbook & Status (local dev)

Last verified: 2025-08-11

This document is the single, simplified reference for the current backend state of ClearKeep: environment, database, migrations, and the green path Parishioner → Sacrament → (Transaction + Calendar).

1) Where things live

- Project root (backend): `C:\ckchurch1\backend`
- Start server: `(venv) C:\ckchurch1\backend> uvicorn app.main:app --host 127.0.0.1 --port 8000 --reload`
- Frontend starter (when used): `C:\ckchurch1\clearkeep-frontend-starter`

2) Environment (.env and Windows env vars)

We use a **sync** SQLAlchemy engine with **psycopg2**. The effective connection string must look like this:

```
DATABASE_URL=postgresql+psycopg2://<user>:<password>@localhost:5432/<database>
```

Windows – recommended setup (persistent)

```
# Set once (User-level) and restart shells after:
setx DATABASE_URL "postgresql+psycopg2://postgres:postgres@localhost:5432/ckchurch"

# Verify in a new PowerShell session:
Get-ChildItem Env:DATABASE_URL
```

Note: We do **not** rely on `.env` being auto-loaded. If you keep a `.env`, treat it as documentation; the running process reads from the OS environment.

Quick diagnostics

```
# What the app sees at startup (logged):
❗ DATABASE_URL = postgresql+psycopg2://...

# If you ever see asyncpg in that line, the driver is wrong.
```

3) Database (PostgreSQL)

Core tables (simplified)

- **parishioners**

- `id` (PK), `first_name`, `last_name`, `contact_number`

- **sacraments**

- `id` (PK)
- `type` (enum: baptism/confirmation/marriage/death)
- `date` (DATE)
- `parishioner_id` (FK → `parishioners.id`)
- `fee` (NUMERIC)
- `details` (JSONB) – flexible fields per type (`child_name`, `time`, `location`, etc.)
- `notes` (TEXT)
- `created_at` (TIMESTAMP)

- **transactions**

- `id` (PK)
- `date`, `description`, `amount`, `type` (income/expense)
- `category_id` (FK → `categories.id`, optional)
- `parishioner_id` (optional)
- `payment_method`, `reference_no`

- **calendar_events**

- `id` (UUID PK)
- `title`, `description`, `location`
- `start_at`, `end_at`, `timezone`
- `rrule`, `exdates[]`
- `origin`, `external_ref`, `meta` (JSONB)
- `is_active` (BOOL)

Invariants (enforced by service layer)

- `Transaction.reference_no` = "SAC-{sacrament.id}"
- `CalendarEvent.external_ref` = "SAC-{sacrament.id}"

- `CalendarEvent.meta` includes `{ sacrament_id, parishioner_id, type }`
- Calendar uses **Asia/Manila** (`ZoneInfo`), default duration **1 hour**
- **No payment info** in calendar description (notes only)

Seed data (categories)

We look up transaction category IDs by name. Ensure these exist:

- Sacraments - Baptism
- Sacraments - Confirmation
- Sacraments - Marriage
- Sacraments - Funeral

(If missing, transactions still save with `category_id = NULL`, but it's better to seed.)

4) Alembic migrations

Location: `C:\ckchurch1\backend` (standard `alembic.ini` + `alembic/` directory).

Common commands:

```
# From C:\ckchurch1\backend (venv active)
# Upgrade DB to latest
alembic upgrade head

# Create a new revision from model changes
alembic revision --autogenerate -m "your message"

# Downgrade one step (if you must)
alembic downgrade -1
```

Tips:

- Ensure `sqlalchemy.url` in `alembic.ini` matches the **psycopg2** URL or have Alembic read from `DATABASE_URL`.
- If autogenerate misses JSONB/Enum nuances, review and edit the generated script.

5) API surface (relevant endpoints)

- `POST /parishioners/` → returns `{ id, ... }`
- `POST /sacraments/` → body `{ parishioner_id, date, fee, notes?, sacrament_type, details? }`
- Side-effects: creates/updates **Transaction** and **CalendarEvent** (ref `SAC-{id}`)

- PATCH /sacraments/{id} → propagates to Tx/Calendar
- DELETE /sacraments/{id} → deletes Tx; deactivates Calendar event
- GET /transactions/?limit=... → ledger
- GET /calendar/events?expand=false → active events only

6) Green-path smoke test (manual)

```
# 1) Create parishioner
$par = Invoke-RestMethod -Method Post -Uri http://127.0.0.1:8000/parishioners/ -
ContentType 'application/json' -Body
(@{first_name='Smoke';last_name='Calendar';contact_number='09999990000'} |
ConvertTo-Json)
$pid = [int]$par.id

# 2) Create sacrament
$body = @{ parishioner_id=$pid; date='2025-08-22'; fee=500; notes='Calendar
smoke'; sacrament_type='baptism'; details=@{child_name='Baby Doe'; time='10:00';
location='Main Church'} } | ConvertTo-Json
$sac = Invoke-RestMethod -Method Post -Uri http://127.0.0.1:8000/sacraments/ -
ContentType 'application/json' -Body $body
$ref = "SAC-$( $sac.id )"

# 3) Verify linkage
$txs = Invoke-RestMethod -Method Get -Uri http://127.0.0.1:8000/transactions/?
limit=1000
$tx = $txs | Where-Object { $_.reference_no -eq $ref } | Select-Object -First 1
$evs = Invoke-RestMethod -Method Get -Uri http://127.0.0.1:8000/calendar/events?
expand=false
$ev = $evs | Where-Object { $_.external_ref -eq $ref } | Select-Object -First 1
```

For a scripted version, see `scripts/smoke.ps1` (part of the Continuity Pack).

7) Start/stop checklist

1. PostgreSQL is running and reachable on `localhost:5432`
2. `DATABASE_URL` shows **psycopg2** (not asyncpg)
3. `alembic upgrade head` ran without errors
4. `uvicorn app.main:app --reload` shows **Application startup complete.**
5. (Optional) `GET /version` shows `db_driver: postgresql+psycopg2`

8) Common pitfalls (and fixes)

- **ForeignKeyViolation on sacraments** → You posted a wrong `parishioner_id`. Confirm the integer you send is from the parishioner you just created.
 - ``` → The URL used the async driver. Switch to `postgresql+psycopg2://...` and restart the shell/server.
 - `*** not applied**` → On Windows, the app doesn't auto-load `.env`. Use `setx` for a persistent env var or set `$env:DATABASE_URL`` in the shell that starts Uvicorn.
 - **PowerShell** `*** confusion**`
→ `$PID` is a built-in variable. Don't reuse that name; use `$parishionerId``.
 - **Duplicate calendar/transactions** on retries → Consider unique index on `transactions.reference_no` and `calendar_events.external_ref`. The service already queries by ref to update, but uniques make it bulletproof.
-

9) What's working today

- Creating a **Sacrament** automatically:
 - Creates **Transaction** (income) with `reference_no = SAC-{id}` and proper category when available
 - Creates **CalendarEvent** (Asia/Manila, +1h) with `external_ref = SAC-{id}` and notes only
 - Updating a sacrament re-syncs Transaction + Calendar
 - Deleting a sacrament deletes the Transaction and deactivates the Calendar event
-

10) Next steps (ops/dev)

- Add unique indexes for the two reference fields
 - Add `/health` (DB ping) and `/version` endpoints
 - Keep `STATUS.md` up to date after each successful smoke test
 - Frontend: use `NEXT_PUBLIC_API_BASE=http://127.0.0.1:8000` and build flows on top of this contract
-

11) Appendix

Verify psycopg2 & SQLAlchemy

```
python - <<'PY'  
import os  
print('DATABASE_URL =', os.getenv('DATABASE_URL'))  
try:  
    import psycopg2  
    print('psycopg2: OK')
```

```
except Exception as e:
    print('psycpg2: FAIL', e)
import sqlalchemy as sa
print('SQLAlchemy:', sa.__version__)
PY
```

Optional indexes (SQL)

```
-- Run once in your DB if desired
CREATE UNIQUE INDEX IF NOT EXISTS ux_transactions_reference_no ON
transactions(reference_no);
CREATE UNIQUE INDEX IF NOT EXISTS ux_calendar_external_ref ON
calendar_events(external_ref) WHERE is_active IS TRUE;
```