

# Procedure Suite

*Automated CPT coding, registry extraction, and synoptic reporting for interventional pulmonology.*

---

## What it is

A FastAPI web UI and API that turns (scrubbed) procedure note text into validated registry data and derived CPT billing codes, with evidence and QA flags.

Current production mode is extraction-first and stateless: text in -> registry + CPT out via POST /api/v1/process.

## Who it's for

Primary users: interventional pulmonology coding/billing specialists and registry coordinators; also clinicians and QA reviewers validating documentation and extracted fields.

## What it does

- Web UI (/ui/) for note paste, PHI detection/redaction workflow, and results review.
- Unified endpoint: POST /api/v1/process (scrubbed text in -> registry + CPT out).
- Extraction-first pipeline: registry extraction (engine via REGISTRY\_EXTRACTION\_ENGINE; recommended parallel\_ner) then deterministic Registry->CPT rules.
- Returns UI-ready JSON with evidence spans and review/status flags.
- Omission scan + RAW-ML auditor; optional guarded self-correction judge (REGISTRY\_SELF\_CORRECT\_ENABLED=1).
- Exports: raw JSON and flattened editable tables (Excel-readable .xls); table edits emit Edited JSON (Training).
- CLI + tests for validation and batch runs (make test, make validate-registry, scripts/registry\_pipeline\_smoke\*.py).

## How it works (repo evidence)

- Client UI: static PHI redactor/dashboard at modules/api/static/phi\_redactor/ served on /ui/.
- Data flow: UI redacts PHI in-browser, then submits scrubbed note text (or sets already\_scrubbed=true).
- API: FastAPI app in modules/api/fastapi\_app.py exposes POST /api/v1/process (modules/api/routes/unified\_process.py).
- Pipeline: run\_unified\_pipeline\_logic -> RegistryService.extract\_fields -> deterministic RegistryRecord -> CPT derivation (CodingService) -> audit/self-correct -> response adapter.
- Key knowledge + schemas: data/knowledge/ip\_coding\_billing\_v3\_0.json; proc\_schemas/registry/; schemas/.

## How to run (minimal)

- Install deps: make install (Python 3.11+).
- Set required env: PROCSUITE\_PIPELINE\_MODE=extraction\_first (service will not start otherwise).
- Configure LLM (optional for some features): GEMINI\_API\_KEY=... (or use offline flags like GEMINI\_OFFLINE=1 / OPENAI\_OFFLINE=1).
- Start: ./scripts/devserver.sh
- Open: <http://localhost:8000/ui/> and <http://localhost:8000/docs>