

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