# 1035 Extractor API - Implementation Guide (Where Code Goes + Exact Steps)

Generated on 2025-09-01

## Repo Layout (put files exactly here)

/app/main.py - FastAPI app■/app/models.py - Pydantic models■/app/parser/preflight.py - OCR and preflight■/app/parser/tables.py - ROI table discovery■/app/parser/mapping.py - Header synonyms and shapes■/app/parser/confidence.py - Confidence blending■/app/parser/pii.py - PII scrub helpers■/app/parser/snips.py - Proof snip cropping■/app/profiles/ - Learned profiles (runtime)■/golden_set/ - Test PDFs■requirements.txt, Dockerfile, .env.example

## Run with Docker (no local setup)

1) docker build -t extractor:latest .■2) docker run --rm -p 8000:8000 extractor:latest■3) Test: curl -F "file=@/path/to/sample.pdf" http://localhost:8000/analyze

## Run without Docker (dev machines)

1) apt-get install: tesseract-ocr ocrmypdf ghostscript qpdf poppler-utils openjdk-17-jre-headless■2) python -m venv .venv && source .venv/bin/activate■3) pip install -r requirements.txt■4) python -m spacy download en_core_web_sm■5) uvicorn app.main:api --host 0.0.0.0 --port 8000

## What your developer must implement next (minimal)

A) Table candidate scoring and scenario selection under Current/Non-Guaranteed■B) Reconciliation math: net_sv approx cash_value - surrender_charge - loan - interest (<=1 percent error)■C) Current-year row selection from issue-year delta or latest complete row■D) PII page redaction boxes and metadata stripping■E) Proof snips (header, first-year, current-year, surrender, loan callout)■F) Auto-profile save and reuse

## Environment variables

CONFIDENCE_THRESHOLD=0.80■PROFILE_DIR=/app/app/profiles■MAX_RUNTIME_SECONDS=60■DISABLE_TABULA=false

## Deploy (example AWS ECS Fargate)

1) Build and push to ECR■2) Create Fargate service (0.5 vCPU, 1GB)■3) Set env vars above■4) Attach ALB on port 8000 and enable HTTPS

## Integration steps (Zapier/Make + OpenAI)

1) Trigger: New PDF in storage■2) HTTP: POST to /analyze with the file■3) OpenAI: send only returned JSON for decision and narrative■4) Doc engine: generate final PDF and embed proof snips

## Acceptance tests

Accuracy >= 95 percent on required fields■Zero PII leaks (JSON, snips, redacted PDF, logs)■Average runtime <= 30s including OCR■Confidence threshold respected; provisional banner when below 0.80

*Note: Do not send raw PDFs to OpenAI. Only scrubbed JSON (and small snips if desired).*