

Release Notes - eDocAPI 0.2.0

Release date: 2026-10-17

Previous version: 0.1.1

Focus: reliability, security, and release quality

Highlights

eDocAPI 0.2.0 focuses on predictable failures, safer uploads, and release reliability. Uploads are validated more strictly, rendering no longer fetches external resources, rate limits protect expensive routes, and CI plus regression tests strengthen the release process.

Security and reliability

- Reject missing uploads, empty document handles, empty converter output, malformed documents, encrypted PDFs, and unsupported operations with predictable framework errors.
- Validate extension, file size, text encoding, and content signatures; reject MIME/signature mismatches and binary content disguised as text.
- Block local and remote resource fetches during HTML, Markdown, and DOCX PDF rendering.
- Stream multipart uploads into managed temporary files under independent per-file and total-request limits.
- Stream file responses in chunks and clean managed files after sending, including interrupted sends.
- Use private permissions for generated temporary files and clean partial conversion outputs on failure.
- Return generic internal errors to clients while keeping diagnostic details in server logs.

Configuration and API

- Apply configured extension allowlists and strict signature checking to uploads.
- Add the independently configurable `max_request_size` limit (100 MiB default).
- Apply OCR enablement and default language configuration to matching OCR handlers.
- Validate every merge input as an unencrypted, readable PDF before merging.
- Add a per-client global rolling-window limiter and per-route overrides; `/health` is exempt.
- Return 429 with `Retry-After` when a client exceeds its configured budget.

Release engineering

- Add CI across supported Python 3.10–3.13 versions, with package build checks and optional-extra installation checks.
- Add regression tests for upload, conversion, rendering, validation, response streaming, and cleanup failures.
- Ignore and remove generated `.pytest-run-*` artifacts.

Document workflow details

- DOCX-to-PDF uses ReportLab's paginated layout engine (core dependency).
- DOCX conversion preserves centered cover titles, heading hierarchy, paragraph order, source table widths, and explicit page breaks.
- Contents pages use readable spacing and indentation for main sections and subsections.
- Tables have visible borders, dark readable text, emphasized first-column labels, and repeating header rows across page breaks.
- Interactive dashboard includes a feature summary and a single `/health` card for both GET and POST.
- Dashboard "Try it" responses, errors, and file previews appear in a modal without leaving the dashboard.
- Structured recognition via optional PaddleOCR (layout, tables, formulas) with `edocapi[structured-ocr]`.

Upload and rate limits

Per-file uploads default to 50 MB (configurable lower per app; framework ceiling is 50 MB). The total request body has an independent `max_request_size` default of 100 MB. Oversized bodies receive 413. Missing or invalid documents receive 400; unsupported types and signature mismatches receive 415. Unexpected server errors return a generic 500 while details stay in logs.

A global rolling-window rate limit per client can be set with `app.limits(...)`, with stricter per-route overrides (e.g. `limit="10/minute"`). The built-in `/health` endpoint is exempt. Over-limit requests receive 429 and a `Retry-After` header.

Credits

Built and maintained by EMMANUEL EMMANUEL ETIM.