

Release Notes - eDocAPI 0.1.1

Release date: 2026-10-15

Previous version: 0.1.0

Highlights

eDocAPI 0.1.1 brings an updated application layer together with practical tools for document processing, conversion, OCR, and secure file handling. This release also improves document layout and gives developers an interactive dashboard for trying API routes.

Breaking changes

- The App class is an ASGI application and can be served with uvicorn, hypercorn, or another ASGI server.
- Import paths remain the same (from `edocapi` import `App`, `Document`), so most user code continues to work.
- Route handlers receive the eDocAPI request interface, available as `edocapi.http.Request`, alongside the documented file and query parameters.

New features

ASGI application

- Built-in request routing, multipart parsing, and response handling for document endpoints.
- Interactive developer dashboard for exploring routes and reviewing results.
- Built-in GET `/health` and POST `/health` checks return the service version and supported formats.

ML-powered document features

- OCR for images and scanned PDFs via Tesseract (`Document(...).ocr()`).
- Optional OpenCV pre-processing (denoising + adaptive thresholding) for higher OCR accuracy.
- Install with: `pip install edocapi[ml]` (and `edocapi[pdf-images]` for PDF page rendering).

Improved converters

- Smarter image compression with iterative quality search toward a target file size.
- Enhanced DOCX → HTML → PDF pipeline (tables, headings preserved).
- Text-to-PDF fallback for reliable basic document rendering.
- Better PDF merge and content-stream compression.

Security hardening

- Strict magic-byte validation (rejects extension spoofing).

- Hardened `safe_filename()` - blocks path traversal, null bytes, and dangerous characters.
- Configurable max file size & max number of files per request.
- Isolated temporary storage with automatic cleanup.
- New `SecurityError` exception for policy violations.
- Bounded multipart body reads, with limits on file counts, form-field counts, and field sizes.
- Kept uploaded files under unique generated names to prevent collisions and overwrite attacks.
- Restricted temporary-file registration and cleanup to the configured storage root.
- Escaped dashboard route metadata and validated route templates to prevent injection.
- Encoded download filenames safely and added response-header validation plus baseline browser security headers.

Packaging & DX

- Clean `pyproject.toml` - `pip install edocapi` just works.
- Optional extras: `html`, `pdf-images`, `ml`, `server`, `all`, `dev`.
- Professional CLI: `edocapi convert|extract|info|ocr|compress`.
- Updated README, examples, and this release note.

Bug fixes & quality

- Consistent error responses (JSON) with proper HTTP status codes (400 / 413 / 500).
- Temporary files are always cleaned up after the response is sent.
- Empty / corrupted files are rejected early with clear messages.
- Logging is configurable via the standard library.

Migration guide (0.1.0 → 0.1.1)

1. Install the new version:

```
pip install -U edocapi
# or with server support
pip install -U edocapi[server]
```

2. Existing `@app.get / @app.post` decorators and `Document` usage remain unchanged.

3. For route type annotations, use `edocapi.http.Request`. Handlers can also accept the documented `file`, `files`, and `query` parameters directly.

Dependencies (core)

Package	Purpose
<code>python-multipart</code>	Multipart form parsing
<code>pypdf</code>	PDF read / write / merge
<code>python-docx</code>	DOCX handling
<code>reportlab</code>	Paginated DOCX-to-PDF layout
<code>markdown</code>	Markdown → HTML

Pillow	Image processing & compression
--------	--------------------------------

Optional: WeasyPrint, pdf2image, pytesseract, opencv-python-headless, uvicorn.

0.1.1 framework refinements (2026-10-15)

- Added a single dashboard entry for /health that supports both GET and POST.
- Added a version and feature summary to the interactive developer dashboard.
- Dashboard endpoint results and errors display in a modal; file responses can be previewed or downloaded without navigating away.
- Improved DOCX PDF typography, contents-page spacing, cover and section page breaks, table borders, text contrast, and repeated table headers.
- Preserved DOCX paragraph order, cover title alignment, heading hierarchy, and source table column widths in PDF output.
- Added ReportLab to core dependencies for paginated DOCX-to-PDF output with formatted headings and tables.

Credits

Built and maintained by EMMANUEL EMMANUEL ETIM.