

You build the customer software. We provide the Document Layer.

DMS, ECM, Document AI, workflows, and 300+ API/MCP tools via REST and MCP — on a single platform from the EU. One access path, two ways: REST for your code, MCP for your AI agents.

Step 1 – Choose your access

Use the access path the product or agent needs. The document remains the same reliable source.

01 · REST API

For your own products

Call Document Operations directly from a backend, workflow or SaaS.

- JSON-based API
- Asynchronous jobs
- Webhooks & events

02 · MCP SERVER

For AI agents

Bring PaperOffice natively into Claude, ChatGPT, Cursor and other MCP clients.

- Tool discovery
- Live document access
- Controlled actions

03 · LLMS.TXT

For coding agents

The coding agent understands capabilities, authentication and examples immediately.

- No SDK required
- Machine-readable docs
- Ready to copy

Step 2 – Create a token

Every product endpoint of the PaperOffice API expects the Authorization: Bearer header. No OAuth flow, no refresh. Both belong on the server. You create, rotate and revoke them in the app under Account → API.

SERVER TO SERVER

Secret Key

- Full access within the account
- Never ship it in browser code
- No Origin header required
- Blocked for MCP

USER-SCOPED

User Token

- Can be limited to individual areas
- Never ship it in browser code
- No Origin header required
- Accepted for MCP

Calls straight from the browser do not use these two tokens but the publishable key `po_pk_` — origin-locked, budget- and rate-limited.

Step 3 – Load the spec and send the first request

SPEC FOR YOUR IDE (LLMS.TXT)

llms.txt for Cursor, Claude Code and ChatGPT.

The PaperOffice API as Markdown following the llmstxt.org standard: index or full spec, ready for your IDE.

<https://api.paperoffice.ai/latest/docs/llms-full.txt>

POSTMAN

The PaperOffice AI API as a Postman collection.

Import one URL, enter the bearer token, send the first request. The overview is built from the live collection.

COOKBOOK

Copy-paste prompts. Ready-made API examples.

28 prompts for document AI, data APIs and IDE workflows — copy, paste, get working code.

Step 4 – Permissions and limits

A user token carries exactly the areas you grant it at creation time. If an area is missing, the API answers with HTTP 403.

Documents Upload, download, processing	documents	Workspaces Manage folders and structure	workspaces
AI Jobs OCR, IDP, extraction	ai_jobs	Billing Read usage and balance	billing
Users Manage team members	users	Webhooks Receive events	webhooks
Knowledge Base Knowledge base and FAQ	knowledge_base	Agents Configure IDP agents	agents
Workflows Create automations	workflows	Compliance Audit, GDPR, archiving	compliance

Counting happens per token; without a bearer, per IP address. The values below are the documented minimums that apply on every plan.

5 requests per second	30 requests per minute	100 requests per hour	500 requests per day
---------------------------------	----------------------------------	---------------------------------	--------------------------------

Remaining budget is in the response header

The RateLimit-* and X-RateLimit-* headers on every response tell you what is left in the current window.

Exceeded: HTTP 429

The API answers with RATE_LIMIT_EXCEEDED. Retry after the time given in the Retry-After header.

Higher plans raise the values

Paid plans go beyond these minimums. Which plan covers what is on the pricing page.

Step 5 – MCP for AI agents

The PaperOffice MCP server brings documents, search, workflows and analytics into Claude, ChatGPT, Grok and Cursor. Remote MCP server — no installation. Paste the URL, OAuth 2.1 or a user token (po_ut_ / po_gt_).

- 1

OAuth 2.1 or po_ut_ / po_gt_
OAuth 2.1 starts on connect.
Alternatively a user or group token (po_ut_ / po_gt_) — not po_sk_.
- 2

Paste the URL into your MCP settings
Pick the URL for your AI client.
- 3

Done — start working
/dms, /cursor and /grok share Documents Operations (300+ MCP tools, including TTS).

Next: webhooks, publishable keys

Webhooks & Events

PaperOffice calls your endpoint as soon as a document, job, workspace or task changes. No polling. 22 event types, HMAC-SHA256 signatures, three retry strategies and a delivery log for every attempt.

Publishable Keys

A publishable key (po_pk_) may live in frontend code: scope-limited, origin-locked and budget-capped. For Claude Artifacts, ChatGPT Canvas, Grok and any web app of your own.

Note. Endpoints, limits and plans are in the live documentation at paperoffice.ai/en/developer/ and in the spec (llms-full.txt); this sheet summarises the first steps (as of October 2026).