

Every invoice different?

PDF in, data out.

Three suppliers, three different layouts, plus a scan with no text. This guide demonstrates how n8n sends an invoice to PaperOffice and retrieves the supplier, invoice number, amount, and due date as JSON. All using standard nodes from n8n, without requiring a custom parser.

TRY IT YOURSELF IN 30 MINUTES

Three invoices, one workflow: retrieve fields in the same response, file documents in your DMS, and receive a signed notification as soon as PaperOffice has finished processing. Includes an import-ready workflow.

WHY THIS GUIDE

One parser per supplier. No one can sustain that.

Teams aiming to automate invoice processing often start by building a custom parser. It works—until the next supplier changes their layout.



Many teams build invoice recognition in-house: a pattern for the invoice number, another for the amount, and a special case for every supplier. Maintenance begins—and never ends—at the first scan without text or when a supplier introduces new letterhead.

This guide demonstrates an alternative approach: n8n sends the document to PaperOffice and receives the extracted fields back as JSON within the same response or as a signed message once PaperOffice has completed processing. You focus solely on orchestrating the workflow, not on building recognition logic. Every step is illustrated with screenshots from n8n and the application. With our test package, you can try it out yourself in just 30 minutes.

Daniel Schönland

Founder and CEO, PaperOffice

WHO THIS IS FOR

Development

You need to integrate invoices into your software without maintaining a custom parser.

IT and Automation

You operate n8n and connect disparate systems.

Accounting

You want to capture vendor, amount, and due date for your payment list without manual data entry.

WHAT YOU CAN ACHIEVE AFTERWARD

Create an API key, set up PaperOffice as an n8n credential, have PaperOffice read the invoices, respond correctly to HTTP 202, file documents in the DMS, subscribe the webhook and verify every delivery by its signature.

OVERVIEW

Contents

01	Three invoices, three layouts	04
02	Why building your own parser is costly	05
03	The process at a glance	06
04	What you need	07
05	In the app: creating an API key	08
06	In n8n: credential and workflow	09
07	Route 1: Send invoice	10
08	The fields as JSON	11
09	Route 2: File in DMS	12
10	Subscribe the webhook	13
11	Verify every delivery	14
12	The row for the payment list	15
13	What teams ask at the start	16
14	Try it yourself in 30 minutes	17
15	Films and more on the topic	18
16	From zero to your first workflow in five days	19
17	Checklist and start	20

AT A GLANCE

- You do not need a dedicated n8n node. Everything runs via standard nodes such as HTTP Request, Webhook, and Code.
- Route 1: n8n sends the invoice and receives the supplier, number, amount, and due date in the same response. If processing takes longer, a small loop handles the polling.
- Route 2: n8n files the invoice in the DMS. Once it has been processed, PaperOffice calls your Webhook.
- Every delivery is signed with HMAC-SHA256. n8n verifies the signature and filters out duplicates.
- The API key lives in an n8n credential, never in the workflow. The API is available in every plan, including Free.

01 · THREE INVOICES, THREE LAYOUTS

The parser holds until the next supplier.

Monday at 9:15 AM. Starting today, incoming invoices for Northstar Industrial Systems in Hamburg should automatically appear in the payment list. The development team is building the workflow in n8n.



Three invoices are on the desk: Kaltwerk Kältetechnik for refrigeration system maintenance, Brandtner Logistik for warehousing and transport, and Metallbau Kröger for the sectional door at Hall 3. Each looks different. The one from Brandtner arrived by mail as a textless scan.

A parser with fixed positions fails at the second invoice. The team would also need text recognition for the scanned invoice.

THE REAL HURDLE

The challenge lies not in the workflow, but in data extraction. PaperOffice handles exactly that. n8n takes care of the rest.

02 · WHY BUILDING YOUR OWN PARSER IS COSTLY

Four reasons. Each new supplier brings a new one.

A parser only recognizes the layouts it has been explicitly trained on.

Every layout is different

The invoice number appears in the top right, bottom left, or subject line. Each new supplier presents a unique challenge.

Scans without text

A paper invoice becomes an image after scanning. Without OCR, the parser cannot extract any data from it.

Numbers in any format

Whether formatted as 1.612,45 EUR, 1612.45, or 1,612.45: each variant requires specific rules before accounting can process it.

Endless maintenance

If a supplier changes their letterhead, the workflow breaks. Often, this isn't noticed until weeks later.

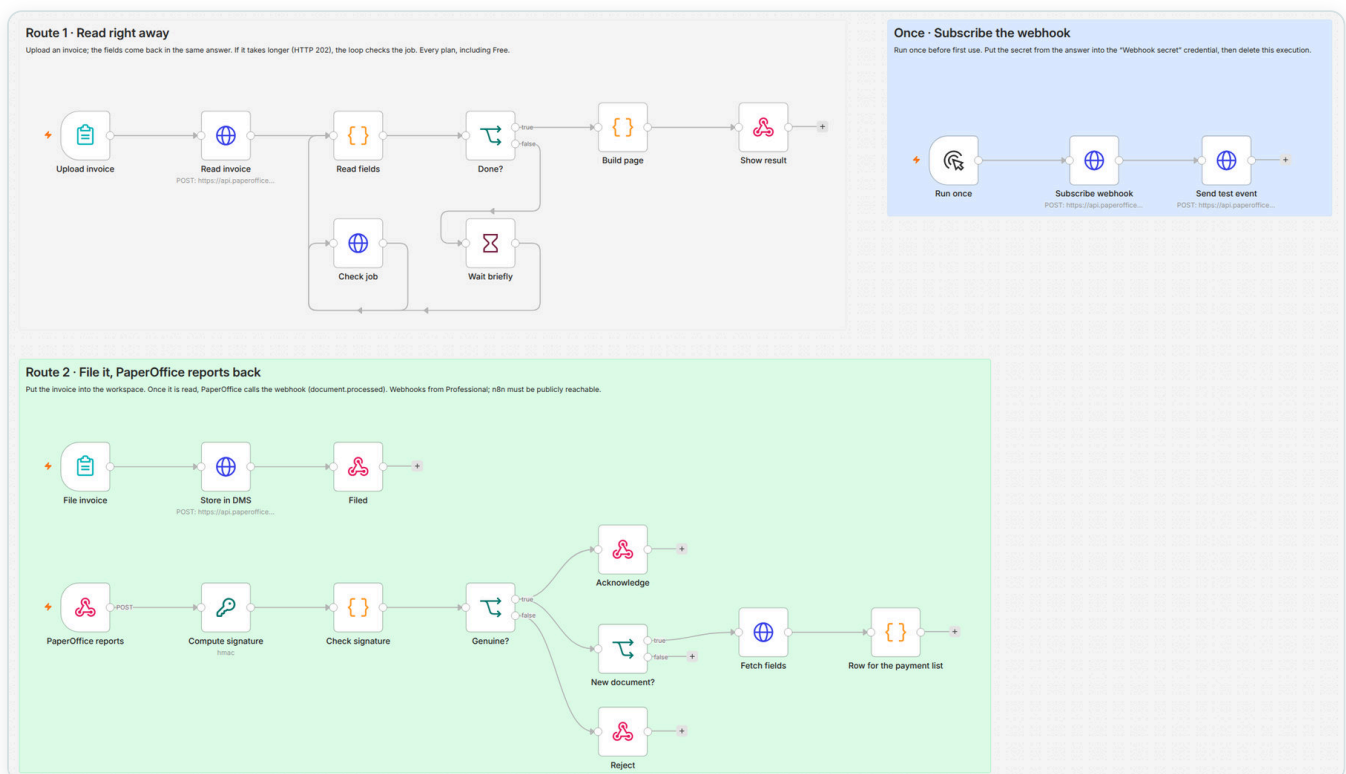
**WHAT CHANGES**

PaperOffice reads the invoice like a human, even as a scan. n8n always receives the same fields, regardless of what the invoice looks like.

03 · THE PROCESS AT A GLANCE

Four steps. Recognition is handled by PaperOffice.

From the key to the line in the payment list. Setup takes one hour; after that, every invoice runs through automatically.



- 1 The workflow from the test package. Top left: Route 1 with the wait loop; right: the one-time subscription of the Webhook; bottom: Route 2 with filing, signature verification, and the line for the payment list. Only standard nodes from n8n.

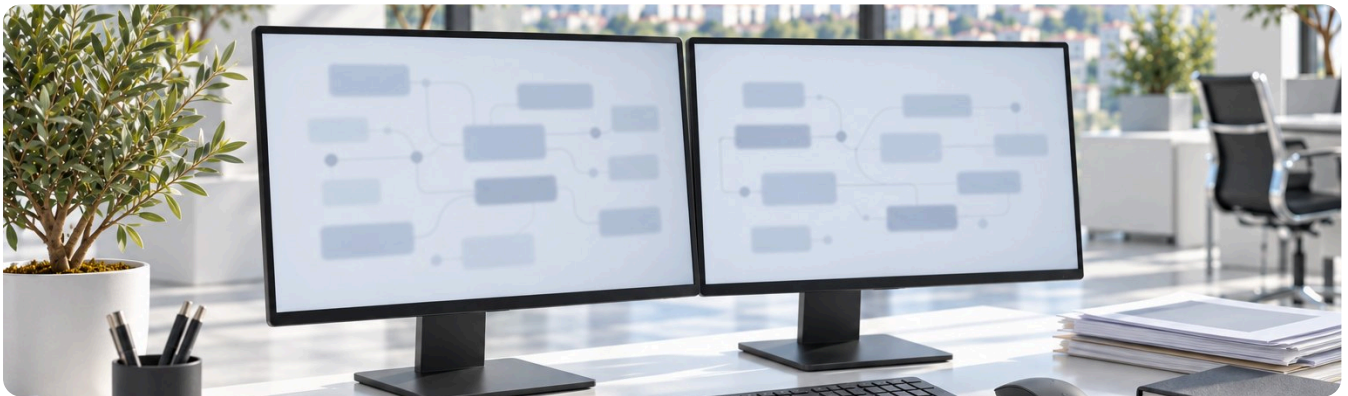
STEP	WHAT PAPEROFFICE DOES	WHAT YOU BUILD IN N8N
Key	validates every call against the API key	a credential of type "Header Auth"
Extract	reads the invoice, even as a scan, and returns the fields; for longer processing times, it first provides a job_id	an HTTP Request node containing the file, along with a brief polling loop
Store	stores the invoice in the Workspace and reads it with AI-DMS	A second HTTP Request node.
Notify	Sends document.processed to your address, digitally signed.	Webhook, signature verification, and your target destination.

04 · WHAT YOU NEED

An account, n8n and, for Webhooks, a public address.

You work with what n8n already includes. You do not need an add-on package or a custom node.

WHAT	WHAT IT LOOKS LIKE
n8n	n8n Cloud or self-hosted. For Route 1, outbound access to api.paperoffice.ai is sufficient.
For Webhooks	Your n8n instance is publicly reachable via http or https. PaperOffice rejects private addresses, localhost, and metadata hosts, and does not follow redirects.
PaperOffice	An account on app.paperoffice.ai. The API is available in every plan, including Free. Webhooks are available from the Professional plan onward.
Key	An API key (poai_...) from "API & Developer" › "API Keys".
Knowledge	Basic understanding of HTTP and JSON. Import the completed workflow from the test package.



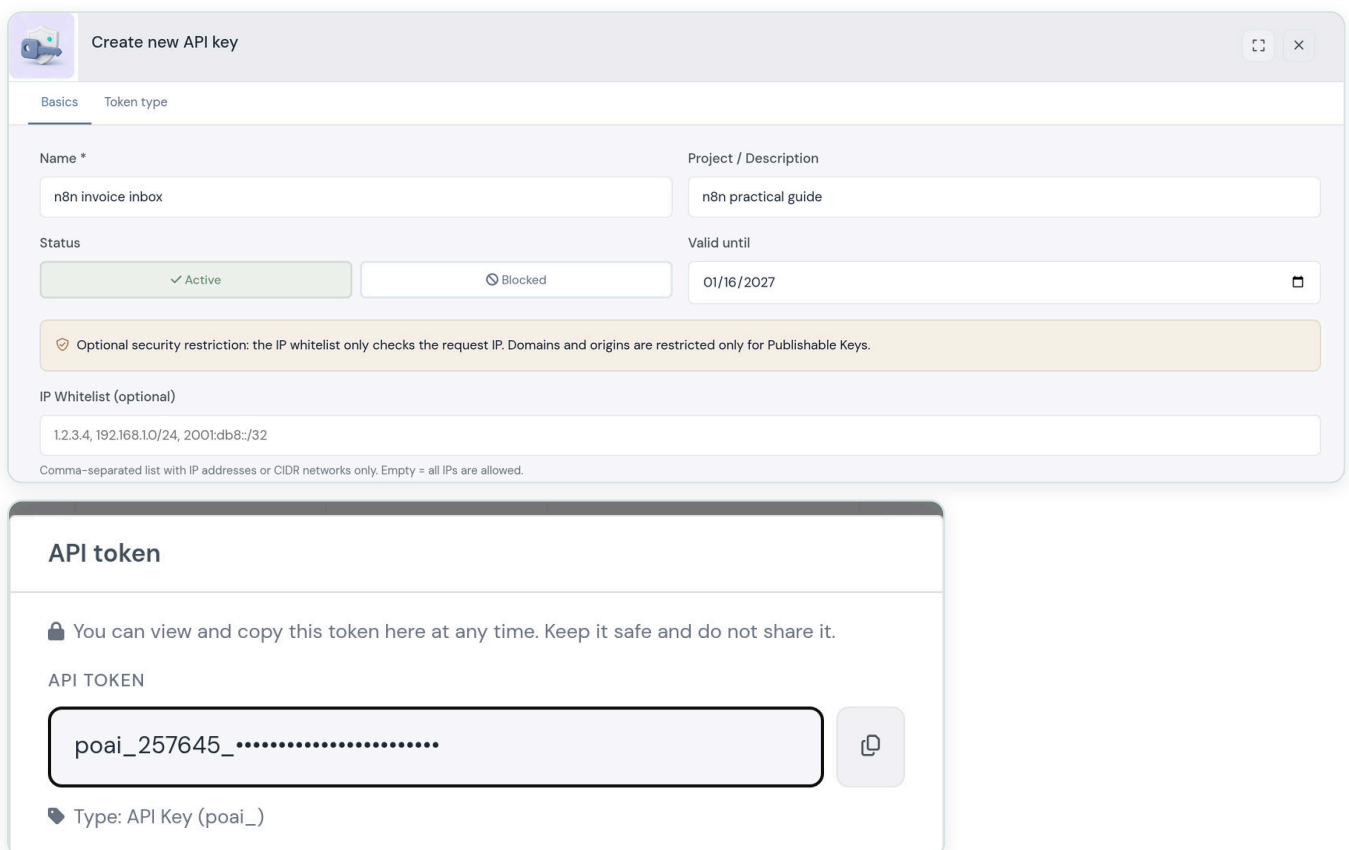
OUR RECOMMENDATION

Start with Route 1. It is available in every plan and demonstrates in just five minutes what PaperOffice extracts from your invoices.

05 · IN THE APP: CREATING AN API KEY

Name, expiration date, type “API Key”. Done.

In the right sidebar under **API & Developer** > **API Keys** , create the key for n8n using “Create New Key”.



Create new API key

Basics Token type

Name * n8n invoice inbox Project / Description n8n practical guide

Status ☒ Active ☐ Blocked Valid until 01/16/2027

Optional security restriction: the IP whitelist only checks the request IP. Domains and origins are restricted only for Publishable Keys.

IP Whitelist (optional)
1.2.3.4, 192.168.1.0/24, 2001:db8::/32

Comma-separated list with IP addresses or CIDR networks only. Empty = all IPs are allowed.

API token

You can view and copy this token here at any time. Keep it safe and do not share it.

API TOKEN

poai_257645_.....

Type: API Key (poai_)

- Create a new API key.** Navigate to the “Basics” tab at the top, where you can define the name and expiration date. The IP whitelist is optional: if you enter the address of your n8n instance there, PaperOffice will accept the key exclusively from that source. Under “Token type”, “API Key” is pre-selected, designed specifically for server-to-server integrations such as n8n. After clicking “Save changes”, PaperOffice displays the key (truncated here). You can view and copy it at any time.

Expiration date

The default validity period is one year. If the key has expired, the API will return an HTTP 401 response along with BEARERTOKEN_EXPIRED.

Store only in Credentials

The key must be stored in the credential vault of n8n, never within a workflow or a shared export.

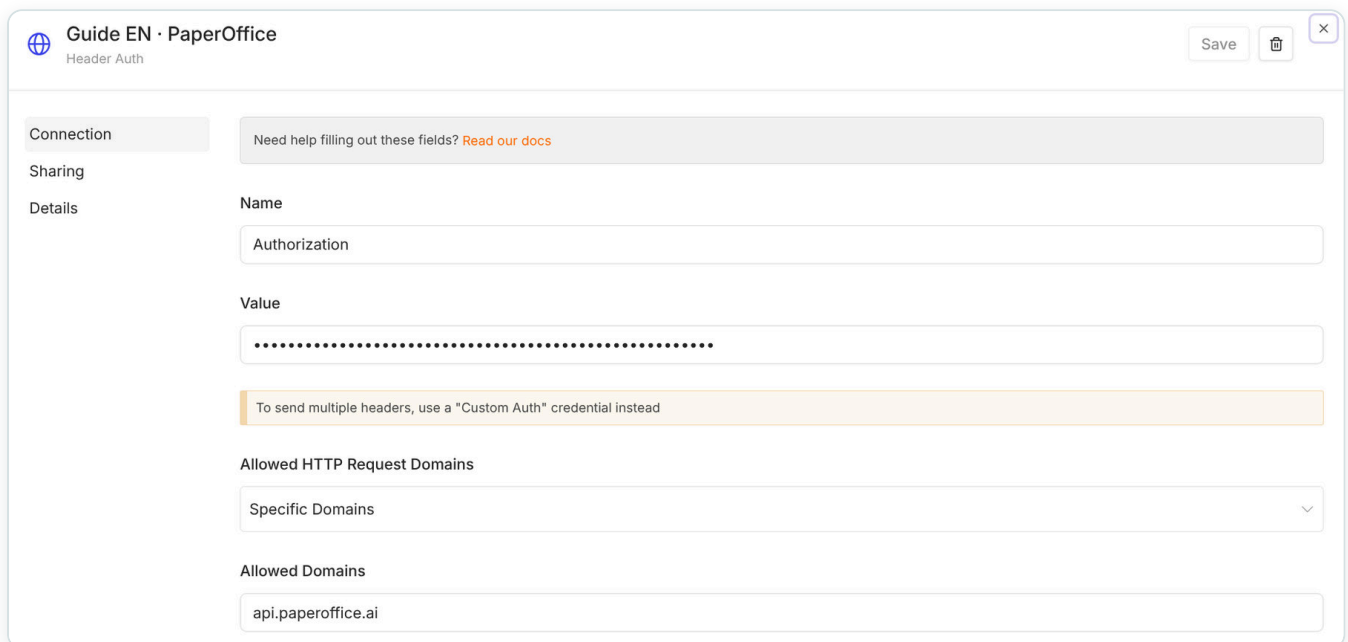
One per workflow

Assigning a dedicated key to each integration allows you to revoke access individually without disrupting other processes.

06 · IN N8N: CREDENTIAL AND WORKFLOW

Set up a credential for the key. Then, import the workflow.

In n8n, create a credential of type "Header Auth". Every HTTP Request node in the workflow will utilize it.



Guide EN · PaperOffice
Header Auth

Need help filling out these fields? [Read our docs](#)

Connection

Sharing

Details

Name
Authorization

Value
.....

To send multiple headers, use a "Custom Auth" credential instead

Allowed HTTP Request Domains
Specific Domains

Allowed Domains
api.paperoffice.ai

- Credential "Header Auth".** Name: "Authorization", Value: "Bearer ", followed by your API key. Under "Allowed HTTP Request Domains", you will find api.paperoffice.ai: n8n ensures the key is sent exclusively to this endpoint, even if someone modifies a node's address.

Import

In a new workflow, open the JSON file from the test package via the "... > "Import" > "From file" menu. It arrives without keys and without secret.

Select Credential

In each HTTP node, select the new credential under "Header Auth", once per node.

Second Credential

For Route 2, add one of type "Crypto" using the Webhook secret. You can find more details in Chapter 10.

07 · ROUTE 1: SEND INVOICE

An HTTP node. The invoice is sent as a file.

Node “Read invoice” sends the file from the form to PaperOffice and waits for the result.

- 1 HTTP Request “Read invoice” in a live run. The invoice arrives from the form on the left. On the right: a POST request to `https://api.paperoffice.ai/latest/job/add/workflow`, authenticated via the Credential. Below that follows the body as Form-Data: `idp_collection = invoice`, `client_wait = true`, and the file under “file”. The Kaltwerk invoice was processed in just under 9 seconds.

RESPONSE	MEANING	HOW THE WORKFLOW RESPONDS
HTTP 200, job_status “completed”	Completed; the fields are available under result	Proceed to “Read fields”
HTTP 202, status “processing”	takes longer; the response contains the job_id	“Wait briefly”, then GET /job/get/{job_id}
job_status “failed”	processing failed, with error_code	Execution terminates with this message.

GOOD TO KNOW

With `client_wait`, PaperOffice maintains the connection until the result is ready, for a maximum of just under five minutes. Consequently, the node's timeout is set to 300 seconds. Queries via `job/get` do not count against rate limits. Whether a job is complete is indicated by `job_status`, not by the HTTP status.

08 · THE FIELDS AS JSON

Supplier, number, amount, due date. From every layout.

The Code node “Read fields” extracts the fields from the response. They are always located in the same position, regardless of the invoice layout.

The screenshot shows the PaperOffice interface with the 'Parameters' and 'Settings' tabs. The 'Mode' is set to 'Run Once for All Items' and the 'Language' is 'JavaScript'. The code node 'Read fields' contains the following JavaScript code:

```
1 // Evaluate the answer of "Read invoice" or "Check job" (fields under
2 const res = $input.first().json || {};
3 let body = res.body !== undefined ? res.body : res;
4 if (typeof body === 'string') { try { body = JSON.parse(body); } catch
(e) { body = {}; } }
```

The 'OUTPUT' tab shows the result of the code execution, which is a JSON object representing the invoice data:

```
{
  "job_id": "poai-job_125_1792356362.83_fc4c70ff337ffaf5e10513c96ce539be",
  "done": true,
  "supplier": "Kaltwerk Kältetechnik GmbH",
  "invoice_number": "KW-2026-1468",
  "total": "1612.45",
  "due_date": "2026-11-13",
  "currency": "EUR",
  "iban": "DE75000000004410001187"
}
```

- 1 Output of “Read fields” for the Kaltwerk invoice. The values are located in the response under result.pages_idp[0].suggested_fields. The Code retrieves “value_raw” where available (date in ISO format, amount as a number); otherwise, it defaults to the display value “value”. If a field is missing, the system returns VALUE_MISSING, which results in an empty value.

The screenshot shows the PaperOffice interface with the 'Invoice to PaperOffice' form on the left and the 'Invoice read' response on the right. The form is titled 'Invoice to PaperOffice' and has a description: 'Upload a PDF or scan of an invoice. PaperOffice reads supplier, number, amount and due date.' The form has a file upload field with the file 'Invoice_KW-2026-1468_Kaltwerk.pdf' and a 'Submit' button. The response is titled 'Invoice read' and shows the following data:

Supplier	Number	Amount	Due date
Kaltwerk Kältetechnik GmbH	KW-2026-1468	1,612.45 EUR	November 13, 2026

- 2 What the sender sees. On the left, the form with the invoice; on the right, the response a few seconds later: A “Respond to Webhook” node sends back the read fields as a small result page.

INVOICE	SUPPLIER	NUMBER	AMOUNT	DUE DATE
Kaltwerk, PDF	Kaltwerk Kältetechnik GmbH	KW-2026-1468	1612.45	2026-11-13
Brandtner, Scan	Brandtner Logistik GmbH	BL-26-04102	1068.62	2026-10-23
Kröger, PDF	Metallbau Kröger GmbH	MK-26-0711	705.67	2026-10-29

09 · ROUTE 2: FILE IN DMS

File the invoice. PaperOffice reads it in the Workspace.

If the invoice is to be stored in the DMS, the “Store in DMS” node sends it to POST /documents/document-put.

Invoice inbox n8n							Options	Current versions	Add document
DOCUMENT	DOCUMENT TYPE	KEYWORDS	DOCUMENT S...	ADDED	AUTOMATION				
 Invoice MK-26-0711 Metallbau Kro... Invoice MK-26-0711 from Metallbau Kröger to Northstar Industrial Systems GmbH This document is an invoice issued by Metallbau Kröger for the maintenance of a sectional door and spare parts supplied to Northstar Industrial System...	Invoice	Metallbau Kröger Northstar Industr... Invoice MK-26-O... sectiona... +4		25 minutes ago 10/18/2026, 09:43 PM	AI-DMS Premium AI-DMS Premium IDP				
 Invoice BL-26-04102 Brandtner Lo... Invoice BL-26-04102 from Brandtner Logistik to Northstar Industrial Systems This is an invoice from Brandtner Logistik for storage and transport services provided in September 2026. The total amount due is EUR 1,068.62, including...	Invoice	Brandtner Logistik Northstar Industr... Invoice BL-26-O... Storage ... +4	Received	25 minutes ago 10/18/2026, 09:42 PM	AI-DMS Premium AI-DMS Premium IDP				
 Invoice KW-2026-1468 Kaltwerk t... Invoice KW-2026-1468 from Kaltwerk Kältetechnik GmbH This document is an invoice issued by Kaltwerk Kältetechnik GmbH to Northstar Industrial Systems GmbH for the maintenance of a refrigeration system i...	Invoice	Kaltwerk Kältetec... Northstar Industr... Refrigeration Syst... Service ... +4		25 minutes ago 10/18/2026, 09:42 PM	AI-DMS Premium AI-DMS Premium IDP				

- The Workspace “Invoice inbox n8n” via Route 2. The three invoices from n8n, processed by AI-DMS Premium: identified as “Invoice”, intelligently named (e.g., “Invoice KW-2026-1468 Kaltwerk to Northstar.pdf”), complete with a summary and relevant keywords.

Name or ID

With workspace_name, PaperOffice uses an existing Workspace or creates one. With workspace_id, it goes directly into this one.

AI-DMS per Workspace

You set how deeply PaperOffice reads on the Workspace. Create it beforehand in the app, with AI-DMS Premium. Amounts are available from Premium.

The response

Under results, each file shows the pofid, the document identifier. You can use it later to retrieve the fields.

ONE OF THE TWO ROUTES IS SUFFICIENT

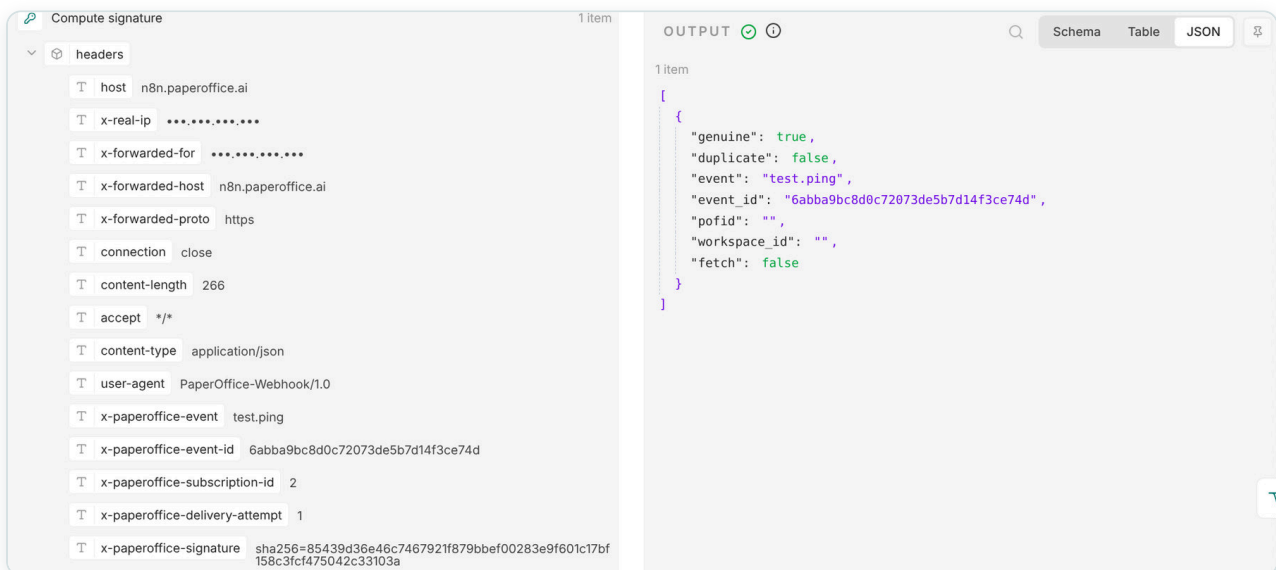
Both Route 1 and Route 2 independently process the invoice. Therefore, please submit the same invoice via only one of these routes, rather than processing it sequentially through both.

10 · SUBSCRIBE THE WEBHOOK

Subscribe once. Then, PaperOffice will report back.

Create Webhook subscriptions via the API, not in the app. The “Once” section of the workflow accomplishes this with two calls.

```
POST https://api.paperoffice.ai/latest/webhooks/subscribe
{
  "name": "n8n invoice inbox",
  "url": "https://YOUR-N8N/webhook/leitfaden-en-verarbeitet",
  "events": ["document.processed"],
  "filters": { "workspace_ids": [22] }
}
// then: POST /webhooks/test with the subscription_id → test.ping
```



The screenshot shows a workflow node titled "Compute signature". On the left, the "headers" section lists various HTTP headers including host, x-real-ip, x-forwarded-for, x-forwarded-host, x-forwarded-proto, connection, content-length, accept, content-type, user-agent, x-paperoffice-event, x-paperoffice-event-id, x-paperoffice-subscription-id, x-paperoffice-delivery-attempt, and x-paperoffice-signature. On the right, the "OUTPUT" section shows a JSON response with fields: genuine (true), duplicate (false), event (test.ping), event_id (6abba9bc8d0c72073de5b7d14f3ce74d), pofid (empty), workspace_id (empty), and fetch (false).

- 1 The test event arrives. On the left, the delivery headers: User-Agent PaperOffice-Webhook/1.0, Event test.ping, Event ID, Subscription ID, Attempt 1, and the signature. On the right "Check signature": real event, no document attached, so no retrieval.

The secret

It appears only in the subscription response; the subscription list never displays it again. Store it in Credential “Webhook secret” (Type Crypto, Field “Hmac Secret”).

The address

The Production URL is located in the Webhook node. It becomes active as soon as the workflow is enabled with “Publish”.

The filter

filters.workspace_ids limits the subscription to Workspace “Invoice inbox n8n”. Other documents do not trigger delivery.

11 · VERIFY EVERY DELIVERY

Real or fake? The signature decides.

Every delivery bears X-PaperOffice-Signature: sha256= and the HMAC-SHA256 over the unaltered raw text, calculated using the secret.

- 1 A genuine delivery of document.processed. Approximately 35 seconds after submission, PaperOffice reports back. The signature is valid (genuine), the event ID is new (duplicate false), so the workflow fetches the fields (fetch).

HEADER	CONTENTS
X-PaperOffice-Event	the event, e.g., document.processed
X-PaperOffice-Event-ID	Unique identifier; remains the same for duplicates
X-PaperOffice-Signature	sha256= followed by the HMAC of the raw payload
X-PaperOffice-Delivery-Attempt	The delivery attempt number

Raw payload

In the Webhook node, "Raw Body" is enabled. The Crypto node computes the HMAC over exactly these bytes, and the secret is retrieved from your credentials.

Duplicates

If no 2xx response is received, PaperOffice retries delivery using the same event ID. The Code node retains the last 500 IDs.

Response

Genuine deliveries are acknowledged by "Acknowledge" with a 200 status code immediately. Invalid signatures receive a 401 response.

12 · THE ROW FOR THE PAYMENT LIST

Retrieve fields. Write the row.

If the delivery is genuine and new, “Fetch fields” fetches the document via GET /documents/document-get/{pofid}. “Row for the payment list” then converts this into a row.

The screenshot shows the PaperoOffice interface with a JavaScript node. The left pane displays the code for mapping document fields to a row. The right pane shows the output, which is a single item containing a JSON object with the following fields: supplier, invoice_number, total, due_date, document, and pofid.

```

1 // Fields from document-get as one row; attach your target here (sheet,
  database, accounting)
2 const res = $input.first().json || {};
3 let body = res.body !== undefined ? res.body : res;
4 if (typeof body === 'string') { try { body = JSON.parse(body); } catch
  (e) { body = {}; } }
5 const d = body.document || {};
6 // field_value_raw is machine-readable (number without extras, amount
  as a number, ISO date), field_value is for display
7 const v = (k) => { const x = d[k]; return x && typeof x === 'object' ?
  String(x.field_value_raw ?? x.field_value ?? '').trim() : String(x ??
  '').trim(); };
8 return [{ json: {
9   supplier: v('_document__data__parties__publisher'),
10  invoice_number: v('_document__data__metadata__document_number'),
11  total: v('_document__data__financial__total_amount'),
12  due_date: v('_document__data__dates__date_due'),
13  document: String(d.file_name || d.filename || d.name || ''),
14  pofid: $('Check signature').first().json.pofid,
15 } }];

```

```

[
  {
    "supplier": "Kaltwerk Kältetechnik GmbH",
    "invoice_number": "KW-2026-1468",
    "total": "1612.45",
    "due_date": "2026-11-13",
    "document": "Invoice_KW-2026-1468_Kaltwerk.pdf",
    "pofid": "727f6f8629fb6acd29b1a00a55f6d69cfe4c1744a746851470f95b689687d0b4.DS50.P005.AI257645.MT1792395098322.pdf"
  }
]

```

- The completed line for the Kaltwerk invoice. On the left, the Code node maps the fields from document-get; on the right, the result: supplier, invoice number, amount, due date, filename, and pofid. Date values are provided in ISO format.

FIELD IN DOCUMENT-GET	IN THE ROW	IN THE TEST
_document__data__parties__publisher	supplier	Kaltwerk Kältetechnik GmbH
_document__data__metadata__document_number	invoice_number	KW-2026-1468
_document__data__financial__total_amount	total	1612.45
_document__data__dates__date_due	due_date	2026-11-13

ATTACH YOUR TARGET

Connect your target to the final node: a table, your database, or accounting software. Since field names remain consistent across all documents, the workflow remains unchanged for future suppliers.

13 · WHAT TEAMS ASK AT THE START

Seven questions before you start.

**Is there a PaperOffice node for n8n?**

No. You will use the standard nodes HTTP Request, Webhook, Code, and Crypto. The completed workflow is included in the test package.

Which key do I need?

For the REST API, you need an API Key (poai_). For AI agents using the MCP Client from n8n, you require a User Token or OAuth; the MCP server does not accept API Keys.

Does n8n need to be publicly accessible?

Yes, for Webhook, via HTTP or HTTPS. For Route 1, n8n only requires outbound access to api.paperoffice.ai.

Are there limits on the number of calls?

At least 5 per second, 30 per minute, 100 per hour and 500 per day apply in every plan; in Free per API key, in paid plans per account. As long as Credits are available, no rate limits apply in paid plans. Queries via job/get do not count.

What happens if n8n is unreachable?

PaperOffice will retry after 30 seconds, then after one, two, four, and eight minutes. The default is five retries, with a maximum of ten. After ten consecutive failures, PaperOffice will deactivate the subscription.

Can this also be done with Make or Zapier?

Yes, using the same pattern: in Make via the HTTP module, and in Zapier via a Custom Request to the REST API.

What does it cost?

Each processing step costs Credits. For exact pricing, please visit our price page: paperoffice.ai/en/pricing.

14 · TRY IT YOURSELF IN 30 MINUTES

Import the workflow. Then extract three invoices.

Three sample invoices from three different suppliers, one of which is a scanned copy, along with the workflow from this guide.

TEST PACKAGE N8N + PAPEROFFICE

Three invoices, the workflow, and an instruction file (README.txt)

Invoice_KW-2026-1468_Kaltwerk.pdf Cooling system maintenance, PDF with text
Invoice_BL-26-04102_Brandtner.pdf Warehouse and transport, scanned
Invoice_MK-26-0711_Kroeger.pdf Sectional door maintenance, PDF with text
workflow_invoice_inbox.json The workflow for importing into n8n
paperoffice.ai/downloads/paperoffice-ai-test-package-n8n.zip



- 1 **Key:** Create an API key at app.paperoffice.ai under "API & Developer" > "API Keys".
- 2 **Workflow:** Import the JSON file into n8n, and set up the credential "Header Auth" with "Bearer " and your key.
- 3 **Route 1:** Activate the workflow with "Publish", open the form "Invoice to PaperOffice", and upload an invoice.
- 4 **Workspace:** for Route 2, create the Workspace "Invoice inbox n8n" in the app with AI-DMS Premium.
- 5 **Webhook:** Enter your address and the Workspace ID in "Subscribe webhook", click "Run once", and input the secret into the credential "Webhook secret".
- 6 **Upload:** upload the three invoices via "File invoice" and review the executions of "PaperOffice reports".

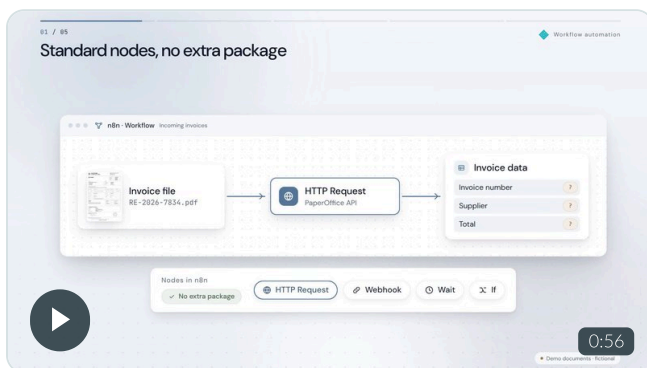
What the test costs is shown on the pricing page: paperoffice.ai/en/pricing

All companies, names, addresses, numbers, and amounts in the test package are fictional.

15 · FILMS AND MORE ON THE TOPIC

Prefer to watch? Two films about n8n.

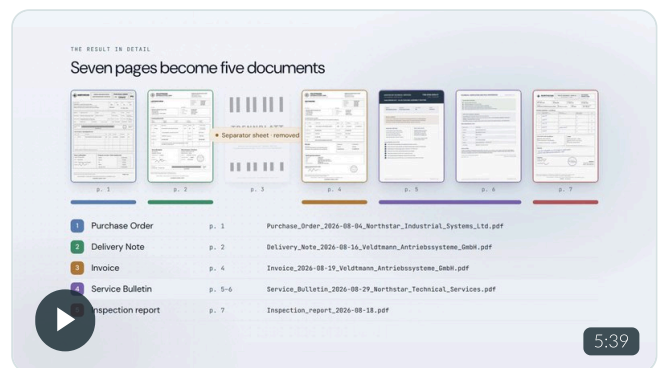
Short films on paperoffice.ai, in English. Scan the QR-Code with your phone camera or open the link.



Connect n8n

REST, Webhook, and MCP in one minute.

paperoffice.ai/en/ai-n8n-integration-workflow-automation



Webinar: Splitting documents with the Split Agent in n8n

A batch scan goes in, individual documents come out.

paperoffice.ai/en/academy/07-integration/webinar-n8n-split-agent



MORE IN THE ACADEMY

You can find all lessons and webinars at paperoffice.ai/l/academy, sorted by topic from the first Workspace to integration.

READ MORE

INTEGRATION

All Integrations

DEVELOPER

Webhooks

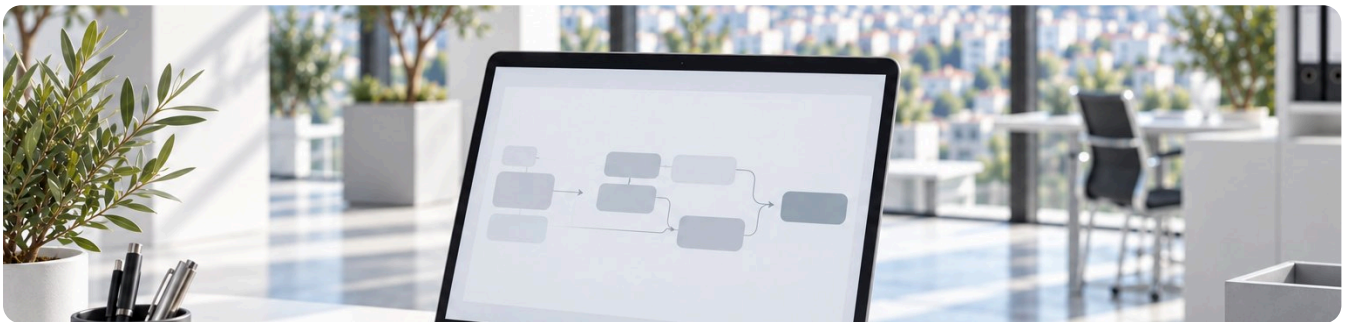
AI AGENTS

MCP Servers

16 · FROM ZERO TO YOUR FIRST WORKFLOW IN FIVE DAYS

Start with ten invoices. Then process every incoming invoice.

To get started, you only need an API key, the workflow from the test package, and ten real invoices from last month.



1

Day 1 · Connect

Create your API key, configure the credential in n8n, import the workflow, and validate it with a test invoice.

2

Day 2 · Extraction

Send ten real invoices via Route 1 and compare the extracted fields with the original documents.

3

Day 3 · Target

Write the fields to your payment list or system of choice.

4

Day 4 · Notification

Set up a Workspace with AI-DMS Premium, subscribe the webhook, turn on signature verification.

5

Day 5 · Operations

Walk through error scenarios: scanned invoices, missing fields, and brief unavailability of n8n.

MEASURE THESE FOUR VALUES
Completeness

Percentage of invoices containing all four fields

Correct

Proportion of fields that required no manual correction

Turnaround Time

From receipt to the line item in the list

Manual Effort

Invoices that still required human intervention

Measure before and after the pilot. Only the comparison reveals how much manual typing is eliminated.

17 · CHECKLIST AND START

Before you expand.

- ☐ A dedicated API key for n8n has been created, complete with an expiration date.
- ☐ The key is stored exclusively within Credential "Header Auth", restricted to api.paperoffice.ai.
- ☐ "Read invoice" has a timeout of 300 seconds; the retry loop captures HTTP 202 responses.
- ☐ The target Workspace has AI-DMS Premium.
- ☐ n8n is publicly accessible for Webhook, and the workflow is active.
- ☐ The secret is stored in the credential "Webhook secret"; the subscription execution has been deleted.
- ☐ Invalid signatures receive a 401 response; duplicates are filtered out.
- ☐ A run with the test package is done.

GET STARTED NOW

Create your account, generate an API key, import the workflow, and extract your first invoice. Start for free with the Free plan—no credit card required.

app.paperoffice.ai**QUESTIONS ABOUT THE GUIDE?**

Please contact us if you have any questions regarding the integration of n8n.

paperoffice.ai/contact

As of October 2026. The screenshots from the app and n8n are taken from test accounts. All examples, company names, personal names, and values in this guide are fictional. n8n is a trademark of n8n GmbH. PaperOffice, paperoffice.ai.