

KOMPLETTANLEITUNG · V2.0

Das komplette MailPilot Handbuch.

Von der Installation bis zum täglichen Betrieb.
IMAP-Sortierung, KI-Klassifikation und Morgen-Digest
in einem n8n-Workflow. Mit und ohne KI nutzbar.

Schritt-für-Schritt-Anleitung mit Beispielen,
Konfigurationsreferenz und Troubleshooting.

4

Nodes

109

Zeilen

21

Features

~350

Mails/Tag

0.02€

KI/Tag

Inhaltsverzeichnis

Kapitel 1 — Was ist MailPilot?

Überblick und Funktionsprinzip

Kapitel 2 — Voraussetzungen

Was du brauchst, bevor es losgeht

Kapitel 3 — Installation Schritt für Schritt

npm-Module, Umgebungsvariablen, n8n-Neustart

Kapitel 4 — Workflow importieren

JSON-Datei laden und Credentials eintragen

Kapitel 5 — Konfigurationsreferenz

Alle Parameter im Detail

Kapitel 6 — Der Workflow im Detail

Was jeder Node macht — Phase für Phase

Kapitel 7 — Die 21 Features

Komplette Feature-Übersicht

Kapitel 8 — Betrieb ohne KI

Wie MailPilot ohne API-Key funktioniert

Kapitel 9 — Anpassungen und Tuning

Zeiten, Filter, Limits individuell einstellen

Kapitel 10 — Datenschutz und DSGVO

Was lokal bleibt, was nach außen geht

Kapitel 11 — Troubleshooting

Die 12 häufigsten Probleme und ihre Lösung

Kapitel 12 — Versionshistorie

Von v7 (22 Nodes) bis v10 (4 Nodes)

Kapitel A — Anhang: Quick-Start-Checkliste

Alle Schritte auf einer Seite

Was ist MailPilot?

Automatische E-Mail-Sortierung und KI-Digest — alles in einem n8n-Workflow.

MailPilot ist ein n8n-Workflow, der drei Aufgaben automatisiert, die sonst täglich manuell erledigt werden müssen:

- 1. Sortierung:** Alle E-Mails älter als 3 Tage werden automatisch in IMAP-Unterordner nach dem Schema `INBOX/domain/absender` verschoben. Fehlende Ordner werden angelegt.
- 2. Spam-Erkennung:** Ein Regex-Vorfilter erkennt Newsletter, noreply-Adressen und Marketing-Absender — noch bevor die KI überhaupt eingesetzt wird.
- 3. KI-Digest:** Die 20 neuesten ungelesenen Mails werden von Claude Haiku klassifiziert (action / info / spam) und als farbcodierter HTML-Digest per E-Mail zugestellt. Jeden Morgen um 7:00 Uhr.

Das Besondere

MailPilot funktioniert **mit und ohne KI**. Ohne Anthropic API-Key sortiert der Workflow trotzdem alle Mails und liefert einen Digest — nur ohne KI-Zusammenfassungen. Alle Daten bleiben auf deinem eigenen Server.

Kennzahlen im Überblick

KENNZAHL	WERT	DETAILS
Nodes	4	Schedule Trigger + 2 Code-Nodes + 1 HTTP Request
Code-Zeilen	109	Gesamter Workflow-Code
Features	21	Alle in v9 und v10 identisch
Max. Mails/Lauf	480	60 Loops x 8 Mails pro Batch
Sortierzeit	< 3 Min	Für ca. 350 Mails
KI-Kosten	~0,02 €/Tag	Bei 20 Mails mit Claude Haiku
Zeitersparnis	~180 h/Jahr	Gegenüber manueller Sortierung

Voraussetzungen

Was du brauchst, bevor du MailPilot einrichten kannst.

2.1 n8n-Instanz

MailPilot läuft auf jeder n8n-Installation ab Version 1.0. Empfohlene Betriebsarten:

VARIANTE	EIGNUNG
Self-Hosted (Proxmox LXC)	Empfohlen — volle Kontrolle, npm-Module direkt installierbar
Docker	Gut geeignet — npm-Module im Container installieren
Bare Metal	Geeignet — npm install direkt im n8n-Verzeichnis
n8n Cloud	Eingeschränkt — keine externen npm-Module, Workaround nötig

2.2 npm-Module

Drei Module werden benötigt:

MODUL	VERSION	FUNKTION
imap	>=0.8.19	IMAP-Verbindung, Mailbox-Operationen, MOVE/Copy+Expunge
mailparser	>=3.6.0	E-Mail-Parsing: MIME-Struktur, Header, Body-Extraktion
nodemailer	>=6.9.0	SMTP-Versand des fertigen Digest

2.3 IMAP/SMTP-Zugangsdaten

Du brauchst folgende Daten von deinem E-Mail-Provider:

FELD	BEISPIEL (IONOS)
IMAP-Host	imap.ionos.de
IMAP-Port	993
SMTP-Host	smtp.ionos.de
SMTP-Port	465
Benutzername	info@meinedomain.de
Passwort	Dein E-Mail-Passwort

Provider-Kompatibilität

MailPilot wurde getestet mit: IONOS, Strato, Hetzner Mail, Gmail (App-Passwort), Outlook/Microsoft 365 und Mailbox.org. Jeder Provider mit Standard-IMAP funktioniert.

2.4 Anthropic API-Key (optional)

Für die KI-Klassifikation brauchst du einen API-Key von Anthropic. Du erhältst ihn unter **console.anthropic.com**. Der Key beginnt mit `sk-ant- . . .`

Kosten: Bei 20 Mails/Tag ca. 0,01–0,02 USD pro Tag (Claude Haiku). Der Spam-Vorfilter reduziert den Token-Verbrauch um ca. 90 %.

Ohne API-Key

Ohne Anthropic-Key funktioniert MailPilot trotzdem vollständig: Sortierung, Ordner-Erstellung und Digest-Versand laufen normal. Nur die KI-Klassifikation und die Zusammenfassungen im Digest entfallen.

Installation Schritt für Schritt

npm-Module installieren, Umgebungsvariable setzen, n8n neu starten.

1 n8n-Verzeichnis finden

Das Verzeichnis hängt von deiner Installation ab. Bei einer Standard-Installation liegt es unter `/usr/lib/node_modules/n8n`. Bei Docker: im Container unter demselben Pfad.

2 npm-Module installieren

Wechsle in das n8n-Verzeichnis und installiere die drei Module:

```
cd /usr/lib/node_modules/n8n
npm install imap mailparser nodemailer --legacy-peer-deps
```

Das Flag `--legacy-peer-deps` verhindert Konflikte mit n8n-internen Abhängigkeiten. Die Installation dauert ca. 10–20 Sekunden.

3 Umgebungsvariable setzen

Damit n8n externe Module in Code-Nodes laden darf, muss folgende Variable gesetzt werden:

```
NODE_FUNCTION_ALLOW_EXTERNAL=*
```

Wo setzen?

METHODE	DATEI / ORT
systemd	<code>`/etc/systemd/system/n8n.service`</code> → <code>Environment=</code>
Proxmox LXC	<code>`/etc/environment`</code> (eine Zeile pro Variable)
Docker	<code>`docker-compose.yml`</code> → <code>environment:</code>
Docker run	<code>-e NODE_FUNCTION_ALLOW_EXTERNAL=*</code>

Beispiel für `/etc/environment`:

```
NODE_FUNCTION_ALLOW_EXTERNAL=*
```

Beispiel für `docker-compose.yml`:

```
environment:
- NODE_FUNCTION_ALLOW_EXTERNAL=*
```

4 n8n neu starten

Nach dem Setzen der Variable muss n8n neu gestartet werden:

```
# systemd:
sudo systemctl restart n8n

# Docker:
docker compose restart n8n
```

5

Installation prüfen

Öffne n8n im Browser. Erstelle einen neuen Workflow mit einem Code-Node und füge folgenden Test-Code ein:

```
const imap = require('imap');
const mp = require('mailparser');
const nm = require('nodemailer');
return [{ json: { status: 'OK', modules: 'loaded' } }];
```

Wenn der Node ohne Fehler läuft und `{status: 'OK'}` zurückgibt, sind alle Module korrekt installiert.

Fehler: Cannot find module

Wenn `Error: Cannot find module 'imap'` erscheint, wurde die Umgebungsvariable nicht gesetzt oder n8n nicht neu gestartet. Prüfe beides und starte erneut.

Workflow importieren

JSON-Datei in n8n laden und die Zugangsdaten eintragen.

1 JSON-Datei herunterladen

Lade die Datei `n8n_imap_sorter_digest_v10.json` herunter oder übertrage sie auf den Server.

2 In n8n importieren

In n8n: oben rechts auf **+** klicken → **Import from File** wählen → die JSON-Datei auswählen. Der Workflow erscheint mit 4 Nodes auf dem Canvas.

3 Code-Node öffnen

Doppelklick auf den Node **Sort+Digest+API**. Im Code findest du oben das Credentials-Objekt:

```
const creds = {
  imapUser: 'DEIN_IMAP_USER',
  imapPass: 'DEIN_PASSWORT',
  imapHost: 'DEIN_IMAP_HOST',
  smtpHost: 'DEIN_SMTP_HOST',
  smtpUser: 'DEIN_IMAP_USER',
  smtpPass: 'DEIN_PASSWORT',
  digestTo: 'DEIN_IMAP_USER',
};
```

4 Platzhalter ersetzen

Ersetze jeden Platzhalter durch deine echten Zugangsdaten:

PLATZHALTER	ERSETZEN DURCH	BEISPIEL
DEIN_IMAP_USER	E-Mail-Adresse / IMAP-Login	info@meinefirma.de
DEIN_PASSWORT	E-Mail-Passwort	MeinSicheresPasswort123
DEIN_IMAP_HOST	IMAP-Server des Providers	imap.ionos.de
DEIN_SMTP_HOST	SMTP-Server des Providers	smtp.ionos.de

5 API-Key eintragen (optional)

Im HTTP-Request-Node **Claude API** den Header `x-api-key` anpassen. Den Platzhalter `DEIN_ANTHROPIC_API_KEY_HIER` durch deinen echten Anthropic API-Key ersetzen. Wenn du keine KI nutzen willst, diesen Node deaktivieren.

6 Testlauf starten

Klicke auf **Test Workflow** (Play-Button). Der erste Lauf erstellt automatisch alle IMAP-Unterordner und kann je nach Postfachgröße 1–5 Minuten dauern.

7

Workflow aktivieren

Wenn der Testlauf erfolgreich war: oben rechts den Toggle auf **Active** setzen. Der Workflow läuft jetzt täglich um 7:00 Uhr automatisch.

Erster Lauf

Beim ersten Start werden alle Mails älter als 3 Tage sortiert. Bei einem vollen Postfach kann das mehrere Minuten dauern. Danach verarbeitet jeder Lauf nur noch die neu eingetroffenen Mails.

Konfigurationsreferenz

Alle einstellbaren Parameter im Code-Node — mit Standardwerten und Erklärung.

PARAMETER	STANDARD	BESCHREIBUNG
maxLoops	60	Max. Durchläufe der Sortierschleife (60 × 8 = 480 Mails)
batchSize	8	Mails pro IMAP-Verbindung (IONOS-Limit)
ageDays	3	Mails jünger als X Tage werden nicht sortiert
digestLimit	20	Max. Mails im Digest
maxBodyLen	800	Body-Text wird auf X Zeichen gekürzt
imapPort	993	IMAP-Port (993 = TLS)
smtpPort	465	SMTP-Port (465 = TLS)
scheduleCron	0 7 * * *	Cron-Ausdruck: täglich 7:00 Uhr
rejectUnauthorized	false	TLS-Zertifikatsprüfung (false = tolerant)
spamRegex	(intern)	Regex für Spam-Vorfilter (noreply, newsletter, marketing)

Wo anpassen?

Alle Parameter stehen im Code-Node **Sort+Digest+API**. Öffne den Node per Doppelklick und suche den entsprechenden Wert im Code. Die meisten Parameter stehen in den ersten 20 Zeilen.

Empfehlung

Die Standardwerte sind für IONOS optimiert. Für andere Provider musst du in der Regel nur Host und Port anpassen. Die Batch-Size von 8 ist ein guter Kompromiss zwischen Geschwindigkeit und Stabilität.

Cron-Beispiele

CRON	BEDEUTUNG
`0 7 * * *`	Täglich um 7:00 Uhr (Standard)
`0 6,18 * * *`	Zweimal täglich: 6:00 und 18:00 Uhr
`*/30 * * * *`	Alle 30 Minuten
`0 7 * * 1-5`	Nur Montag bis Freitag um 7:00 Uhr

Der Workflow im Detail

Was jeder Node macht — Phase für Phase erklärt.

Übersicht: 4 Nodes, linearer Ablauf

NODE	TYP	AUFGABE
7 Uhr	Schedule Trigger	Startet den Workflow nach Cron-Zeitplan
Sort+Digest+API	Code (JS)	Phase 1–3: Sortierung, Digest sammeln, API-Request
Claude API	HTTP Request	POST an api.anthropic.com (KI-Klassifikation)
Parse+HTML+SMT P	Code (JS)	KI-Antwort parsen, HTML-Digest bauen, SMTP-Versand

Phase 1 — Sortierung (While-Loop)

Der Code-Node öffnet eine IMAP-Verbindung und liest bis zu 8 Mails aus INBOX. Für jede Mail wird der Absender extrahiert und ein Zielordner nach dem Schema `INBOX/domain.tld/absender@domain.tld` bestimmt. Fehlende Ordner werden per `imap.addBox()` angelegt (mit `ALREADYEXISTS`-Handling). Die Mail wird per IMAP MOVE verschoben — falls der Provider MOVE nicht unterstützt, greift ein Fallback aus Copy + Expunge.

Nach jedem 8er-Batch wird die Verbindung geschlossen und neu aufgebaut. Das verhindert Timeout-Probleme bei IONOS und anderen Providern. Die Schleife läuft maximal 60 Mal (= 480 Mails).

Phase 2 — Digest sammeln (ReadOnly)

Eine separate IMAP-Verbindung (ReadOnly) lädt die 20 neuesten ungelesenen Mails. Jede Mail wird durch den Spam-Vorfilter geschickt — ein Regex prüft auf Muster wie `noreply`, `newsletter`, `marketing`, `unsubscribe`. Erkannte Spam-Mails erhalten einen leeren Body. Das spart ca. 90 % der API-Token, weil der KI nur der Betreff übergeben wird.

Phase 3 — KI-Klassifikation

Der Code-Node baut einen JSON-Body für den HTTP-Request an Claude Haiku. Das Prompt enthält die Mail-Liste mit Betreff und gekürztem Body (max. 800 Zeichen). Claude klassifiziert jede Mail als:

KATEGORIE	FARBE	BEDEUTUNG
action	Rot	Handlungsbedarf — muss bearbeitet werden
info	Blau	Zur Kenntnis — keine Aktion nötig
spam	Grau	Spam / Newsletter / irrelevant

Phase 4 — Digest und Versand

Der letzte Code-Node parst die KI-Antwort (JSON) und ordnet die Klassifikation den Mails zu. Daraus wird ein HTML-Digest gebaut: action-Mails stehen oben (rot), dann info (blau), dann spam (grau). Ein Akkordeon-Anhang (HTML details/summary) ermöglicht das Aufklappen einzelner Mails. Der Digest wird per nodemailer/SMTP an die konfigurierte Adresse versendet.

Die 21 Features

Was MailPilot alles kann — mit und ohne KI.

Sortierung und IMAP

#	FEATURE	DETAILS
1	Auto-Sortierung	Mails nach Domain/Absender in IMAP-Unterverordner verschoben
2	While-Loop	Interne Schleife statt n8n-Loop-Node (performanter)
3	Batch-Size 8	Max. 8 Operationen pro IMAP-Verbindung (Provider-stabil)
4	Max 60 Loops	Bis zu 480 Mails pro Durchlauf verarbeitbar
5	Age-Filter	Nur Mails älter als 3 Tage werden sortiert
6	IMAP MOVE	MOVE-Befehl mit Copy+Delete als Fallback
7	Auto-Ordner	Fehlende Ordner werden angelegt (ALREADYEXISTS-Handling)
8	Reconnect	Neue IMAP-Verbindung nach jedem Batch

Spam und KI

#	FEATURE	DETAILS
9	Spam-Vorfilter	Regex erkennt Newsletter/Marketing vor der KI
10	Spam ohne Body	Spam-Mails mit leerem Body (90 % Token-Ersparnis)
11	KI-Klassifikation	Claude Haiku: action / info / spam pro Mail
12	Ohne KI nutzbar	Sortierung + Digest auch ohne API-Key

Digest und Versand

#	FEATURE	DETAILS
13	Prio-Sortierung	Digest sortiert nach Dringlichkeit (action zuerst)
14	HTML-Body	Farbcodiert: rot (action), blau (info), grau (spam)
15	Akkordeon-Anhang	details/summary-HTML zum Aufklappen einzelner Mails
16	SMTP-Versand	Digest per nodemailer mit TLS
17	TLS-kompatibel	rejectUnauthorized für alle Provider
18	Digest-Limit 20	Max. 20 Mails im Digest (konfigurierbar)

Technisch

#	FEATURE	DETAILS
19	Max Body 800	Body auf 800 Zeichen gekürzt (Token-Optimierung)
20	onError	continueRegularOutput für Fehlertoleranz
21	Mailparser	MIME-Parsing für korrekte Header- und Body-Extraktion

Betrieb ohne KI

Wie MailPilot funktioniert, wenn du keinen API-Key verwendest.

MailPilot ist so gebaut, dass die KI-Klassifikation optional ist. Wenn du keinen Anthropic API-Key einträgst oder den Claude-API-Node deaktivierst, passiert Folgendes:

FUNKTION	MIT KI	OHNE KI
Sortierung in Unterordner	Ja	Ja (identisch)
Ordner automatisch anlegen	Ja	Ja (identisch)
Spam-Vorfilter (Regex)	Ja	Ja (identisch)
Digest-Versand per SMTP	Ja	Ja (ohne KI-Zusammenfassung)
KI-Klassifikation	action/info/spam	Nicht verfügbar
Zusammenfassung pro Mail	1 Satz pro Mail	Nicht verfügbar
Farbcodierung im Digest	Rot/Blau/Grau	Einheitlich
API-Kosten	~0,02 €/Tag	0,00 €

So deaktivierst du die KI:

Rechtsklick auf den Node **Claude API** → **Deactivate**. Der Workflow überspringt den Node und der letzte Code-Node erkennt automatisch, dass keine KI-Antwort vorliegt.

Jederzeit umschaltbar

Du kannst die KI jederzeit wieder aktivieren, indem du den Node einschaltest und einen gültigen API-Key einträgst. Es ist kein Umbau am Workflow nötig.

Anpassungen und Tuning

Zeiten, Filter, Limits und Verhalten individuell einstellen.

Sortier-Zeitpunkt ändern

Im Schedule-Trigger-Node den Cron-Ausdruck anpassen. Beispiel für zweimal täglich (6:00 und 18:00):

```
0 6,18 * * *
```

Age-Filter anpassen

Standardmäßig werden nur Mails älter als 3 Tage sortiert. Im Code-Node die Variable `ageDays` ändern. Wert 1 sortiert Mails ab 24 Stunden.

Digest-Limit erhöhen

Die Variable `digestLimit` bestimmt, wie viele Mails maximal im Digest erscheinen. Standard: 20. Bei mehr Mails steigen die API-Kosten proportional.

Spam-Vorfilter erweitern

Der Regex-Spam-Filter steht im Code-Node. Um weitere Muster hinzuzufügen, den Regex erweitern. Beispiel — zusätzlich `promotion` als Spam erkennen:

```
/(noreply|newsletter|marketing|unsubscribe|promotion)/i
```

Mehrere Postfächer

Für jedes zusätzliche Postfach den Workflow duplizieren und mit eigenen Zugangsdaten versehen. Jede Instanz läuft unabhängig. Die Cron-Zeiten leicht versetzt setzen (z. B. 7:00, 7:05, 7:10), um gleichzeitige API-Calls zu vermeiden.

Digest-Empfänger ändern

Im Credentials-Objekt den Wert `digestTo` auf eine andere E-Mail-Adresse setzen. Der Digest kann auch an eine Verteiler-Adresse gehen.

Datenschutz und DSGVO

Was lokal bleibt, was nach außen geht — und warum das DSGVO-konform ist.

Datenfluss-Übersicht

DATEN	VERBLEIB
Alle E-Mails (komplett)	Bleiben auf deinem IMAP-Server — werden nie kopiert
Anhänge	Werden nicht verarbeitet — bleiben unberührt
IMAP-Zugangsdaten	Nur im n8n Code-Node gespeichert (lokal)
Sortierte Ordnerstruktur	Liegt auf deinem IMAP-Server
Betreff + Body (gekürzt)	Wird an Anthropic API gesendet (nur mit KI)
API-Key	Wird an api.anthropic.com gesendet (HTTPS)
Digest-HTML	Wird per SMTP an dein Postfach gesendet

Was geht an Anthropic?

Nur bei aktivierter KI-Klassifikation: Betreff und die ersten 800 Zeichen des Body-Texts. Keine Anhänge, keine Empfänger-Adressen, keine Metadaten. Spam-Mails werden mit leerem Body übergeben (nur Betreff).

Anthropic speichert laut eigener API-Policy keine Eingabedaten für Training. Die Übertragung erfolgt verschlüsselt über HTTPS.

Zusammenfassung

Deine E-Mails verlassen deinen Server nicht. An Anthropic geht nur ein minimaler Auszug (Betreff + gekürzte Body-Texte) — und auch das ist komplett abschaltbar. Ohne KI = null externe Kommunikation.

Troubleshooting

Die 12 häufigsten Probleme — mit Ursache und Lösung.

1. „Cannot find module 'imap'“

Ursache: NODE_FUNCTION_ALLOW_EXTERNAL nicht gesetzt oder n8n nicht neu gestartet.

Lösung: Variable in /etc/environment oder systemd setzen, n8n restarten.

2. „Auth-Fehler am Claude-API-Node“

Ursache: API-Key-Platzhalter wurde nicht ersetzt.

Lösung: Im HTTP-Request-Node den Header x-api-key prüfen und echten Key eintragen.

3. „Mails werden nicht sortiert“

Ursache: Mails sind jünger als 3 Tage (Age-Filter).

Lösung: Warten oder ageDays auf 1 setzen. Auch IMAP-Zugangsdaten prüfen.

4. „Digest kommt nicht an“

Ursache: SMTP-Zugangsdaten falsch oder Spam-Ordner des Empfängers.

Lösung: SMTP-Host/Port/Passwort prüfen. Viele Provider brauchen Port 465 mit TLS.

5. „Timeout bei großem Postfach“

Ursache: Erste Sortierung bei >500 Mails dauert lange.

Lösung: Normal beim ersten Lauf. Reduziere maxLoops auf 30 für den Anfang.

6. „IMAP MOVE schlägt fehl“

Ursache: Provider unterstützt MOVE nicht.

Lösung: MailPilot nutzt automatisch Copy+Expunge als Fallback.

7. „Ordner werden nicht angelegt“

Ursache: IMAP-Berechtigungen eingeschränkt.

Lösung: Prüfe, ob dein Provider das Anlegen von Unterordnern erlaubt.

8. „API-Kosten zu hoch“

Ursache: digestLimit zu hoch oder Spam-Vorfilter greift nicht.

Lösung: digestLimit auf 15 senken, Spam-Regex erweitern.

9. „Workflow startet nicht automatisch“

Ursache: Workflow nicht aktiviert (Toggle oben rechts).

Lösung: Toggle auf Active setzen. Prüfe auch den Cron-Ausdruck.

10. „HTML-Digest zeigt keine Farben“

Ursache: E-Mail-Client blockiert Inline-CSS.

Lösung: Normal bei einigen Clients. Outlook und Gmail zeigen die Farben korrekt.

11. „Doppelte Mails in Unterordnern“

Ursache: Copy+Expunge-Fallback bei Verbindungsabbruch vor Expunge.

Lösung: Tritt selten auf. Duplikate manuell löschen, beim nächsten Lauf OK.

12. „n8n Task Runner Crash“

Ursache: Versuch, require('axios') oder require('https') im Code-Node zu nutzen.

Lösung: Diese Module werden von n8n blockiert. Daher läuft der API-Call über den HTTP-Request-Node.

Versionshistorie

Von v7 mit 22 Nodes bis v10 mit 4 — bei gleichem Funktionsumfang.

VERSION	NODES	ZEILEN	NEUERUNGEN
v7	22	~450	Erster produktiver Workflow: jede Funktion ein eigener Node, n8n-Loop-Node
v8	14	~320	Nodes zusammengelegt, Code konsolidiert, weniger Canvas-Verbindungen
v9	7	~180	While-Schleife statt Loop-Node, Spam-Vorfilter, Mega-Code-Nodes
v10	4	109	Absolute Komprimierung: kein Manual Trigger, keine Sticky Notes

Alle vier Versionen bieten denselben Funktionsumfang (21 Features). Die Komprimierung betrifft ausschließlich die Code-Architektur.

Warum nicht 3 Nodes?

n8n blockiert HTTP-Requests aus Code-Nodes (Sandbox). `require('https')`, `require('http')` und `require('axios')` verursachen Crashes. Der Claude-API-Call muss daher über einen nativen HTTP-Request-Node laufen. 4 Nodes ist die absolute technische Untergrenze.

Quick-Start-Checkliste

Alle Schritte auf einen Blick — zum Abhaken.

01 ■ n8n-Instanz

n8n ist installiert und erreichbar (Self-Hosted oder Docker)

02 ■ npm install

```
cd /usr/lib/node_modules/n8n && npm install imap mailparser nodemailer --legacy-peer-deps
```

03 ■ Umgebungsvariable

NODE_FUNCTION_ALLOW_EXTERNAL=* gesetzt

04 ■ n8n Neustart

systemctl restart n8n oder docker compose restart n8n

05 ■ Modul-Test

Test-Code-Node mit require('imap') gibt OK zurück

06 ■ Workflow importieren

n8n_imap_sorter_digest_v10.json geladen

07 ■ IMAP-Credentials

User, Passwort, Host im Code-Node eingetragen

08 ■ SMTP-Credentials

Host, Port, Passwort eingetragen

09 ■ API-Key (optional)

Anthropic Key im HTTP-Request-Node Header eingetragen

10 ■ Testlauf

Test Workflow ausgeführt — Ordner werden angelegt

11 ■ Digest prüfen

E-Mail mit Digest erhalten und geprüft

12 ■ Workflow aktivieren

Toggle auf Active — läuft jetzt automatisch