

Intelligentes Materialmapping

Freitext per Telegram, WhatsApp oder Threema wird
per KI zum Lexware-Angebotsentwurf — nur echte Artikelnummern.

Version	1.1
Stand	16.08.2026
Status	Einsatzbereit — Mehrkanal (3 Messenger rein, E-Mail + 3 Messenger raus)
System	n8n-Workflow · Telegram · WhatsApp · Threema · E-Mail · KI · Lexware Office API · 35 Nodes
Herausgeber	FlowTrix · flowtrix.de

Inhaltsverzeichnis

1 · Was ist das System?	3
2 · Voraussetzungen	3
3 · Erste Schritte	4
4 · Benutzer und Berechtigungen	4
5 · Hauptfunktionen	5
6 · Typischer Arbeitsablauf	6
7 · Die Ansicht	6
8 · Administration	7
9 · Datenschutz und Sicherheit	8
10 · Fehlerbehebung	8
11 · Häufige Fragen & Einschränkungen	9
12 · Support und Versionshistorie	9

Was ist das System?

Ein n8n-Workflow, der eine formlose Materialangabe („3 Fenster, 2 Türen“) per KI auf den Lexware-Artikelstamm mappt und daraus einen **Angebotsentwurf** anlegt (`POST /quotations` , ohne Finalisierung → Status Entwurf). Nicht zuordenbare Angaben werden als 0-€-Platzhalter „PRÜFEN“ abgelegt und zusätzlich gesondert gemeldet.

Mehrkanal. Eingang über drei Messenger als Freitext: Telegram, WhatsApp und Threema. **Ausgabe** von Ergebnis und Fehler über E-Mail (direkt) sowie Telegram, WhatsApp und Threema über den Messenger-Baustein. Ein leer gelassenes Ziel wird übersprungen.

Nur echte Artikelnummern. Die KI darf ausschließlich Artikelnummern aus dem geladenen Stamm verwenden. Der Node `req: Angebot anlegen` prüft zusätzlich gegen den Katalog-Index — unbekannte oder `null` -Nummern werden nie als echte Position übernommen.

Voraussetzungen

- Laufende **n8n-Instanz**
- **Lexware-Office-API-Baustein** eingerichtet (Auth, Basis-URL, Rate-Limit)
- **Messenger-Baustein** eingerichtet (Telegram/WhatsApp/Threema) für den Versand
- Eingangs-Credentials: **Telegram-Bot, WhatsApp-Trigger, Threema-Gateway** (je nach genutzten Kanälen)
- **SMTP-Zugang** für die E-Mail-Ausgabe
- Zugang zu einem **OpenAI-kompatiblen Chat-Endpunkt** (Cloud oder lokal)

Erste Schritte

1. Workflow-Datei in n8n importieren.
2. In den drei Lexware-Baustein-Nodes den Baustein-Workflow neu auswählen: „**Baustein: Bestand Seite 0**“, „**Baustein: weitere Seite**“, „**Baustein: Angebot anlegen**“.
3. Im Node „**Baustein: Nachricht senden**“ den „**Messenger – Nachricht senden (Baustein)**“ wählen (Mode „Run once for each item“).
4. Im Node „**KI: Mapping**“ ein **eigenes Credential** (Header Auth, `Authorization: Bearer <Key>`) setzen — nicht über den Baustein.
5. Eingangs-Credentials setzen: **Telegram-Bot** (Trigger), **WhatsApp-Trigger**, **Threema-Gateway** (Callback-URL im Gateway hinterlegen).
6. Im Node „**Mail: Ausgabe**“ SMTP-Credential und Absender setzen.
7. In **Config** die Ausgabe-Ziele eintragen: `chefEmail`, `absenderEmail`, `telegramChatId`, `whatsappZiel`, `threemaZiel`; bei Bedarf `aiBaseUrl`, `aiModel`, `seitengroesse` anpassen.
8. **Workflow aktiv schalten**.
9. Eine Materialangabe an einen der Messenger schicken und das Ergebnis prüfen.

Telegram-Chat-ID herausfinden: dem Bot `@userinfobot` in Telegram schreiben — er nennt die eigene Chat-ID.

Leeres Ziel = Kanal übersprungen. Es müssen nicht alle Kanäle gesetzt sein — der Node `Ausgabe-Ziele bauen` bedient nur gefüllte Ziele. So lässt sich der Versand kanalweise zu- oder abschalten.

Benutzer und Berechtigungen

Rolle	Aufgabe	Braucht
Absender (Kalkulation)	Materialangabe per Messenger schicken	Telegram / WhatsApp / Threema
Empfänger	erhält Ergebnis- und PRÜFEN-Meldung	E-Mail und/oder Messenger
Administrator	richtet Bausteine, KI-Credential, SMTP, Messenger-Ziele ein	n8n-Zugang

Hauptfunktionen

5.1 Freitext-Empfang über drei Messenger

Die Materialangabe kann als formloser Text über **Telegram**, **WhatsApp** oder **Threema** eingehen. Jeder Kanal hat einen eigenen Trigger und eine Normalisierung, die den Text auf ein einheitliches Feld `freitext` bringt. Der Node `Freitext holen` übernimmt den Text kanal-unabhängig; ein leerer Text stoppt den Lauf sauber.

5.2 Artikelstamm laden

Der Lexware-Artikelstamm wird paginiert über den Baustein geladen (`Bestand Seite 0` → weitere Seiten über `splitInBatches`). Archivierte Artikel werden ausgefiltert; übrig bleibt eine kompakte Liste `{articleNumber, title, unitName, netPrice}`.

5.3 KI-Mapping

Der Node `KI: Mapping` schickt System-Prompt, Freitext und Artikelliste an einen OpenAI-kompatiblen Endpunkt. Die KI antwortet als striktes JSON-Array `[{eingabe, articleNumber, menge}]`; `temperature: 0` sorgt für reproduzierbare Zuordnung.

5.4 Katalog-Prüfung und PRÜFEN-Markierung

Der Node `req: Angebot anlegen` prüft jede von der KI genannte Nummer gegen den Katalog-Index. Unbekannte oder `null`-Nummern werden nie als echte Position übernommen, sondern als 0-€-Platzhalter „PRÜFEN“ abgelegt und in der Ergebnismeldung gesondert aufgeführt.

5.5 Angebotsentwurf

Aus den Treffern entsteht ein Angebot (`POST /quotations`) **ohne Finalisierung** — Status Entwurf. Positionen als `type: custom` mit dem Katalog-Netto-Preis; Adresse als Platzhalter „Kunde bitte zuordnen“.

5.6 Ausgabe: E-Mail + drei Messenger

Ergebnis **und** Fehler werden auf allen konfigurierten Kanälen gemeldet: per **E-Mail** direkt (`Mail: Ausgabe`, SMTP) und über **Telegram**, **WhatsApp** und **Threema** mittels des **Messenger-Bausteins**. Der Node `Ausgabe-Ziele bauen` erzeugt je gesetztem Ziel ein Item; ein leeres Ziel wird übersprungen. Die Ausgabe-Nodes laufen mit `continueRegularOutput` — ein ausgefallener Kanal stoppt die übrigen nicht.

Typischer Arbeitsablauf

1. Absender schickt eine formlose Materialangabe per Telegram, WhatsApp oder Threema.
2. Der Workflow lädt den Lexware-Artikelstamm und reduziert ihn auf das Nötige.
3. Die KI ordnet die Angaben existierenden Artikelnummern zu.
4. Jede Nummer wird gegen den Katalog geprüft; Unklares wird „PRÜFEN“.
5. Ein Angebotsentwurf wird angelegt und das Ergebnis (inkl. PRÜFEN-Liste) über E-Mail und alle konfigurierten Messenger gemeldet.

Die Ansicht

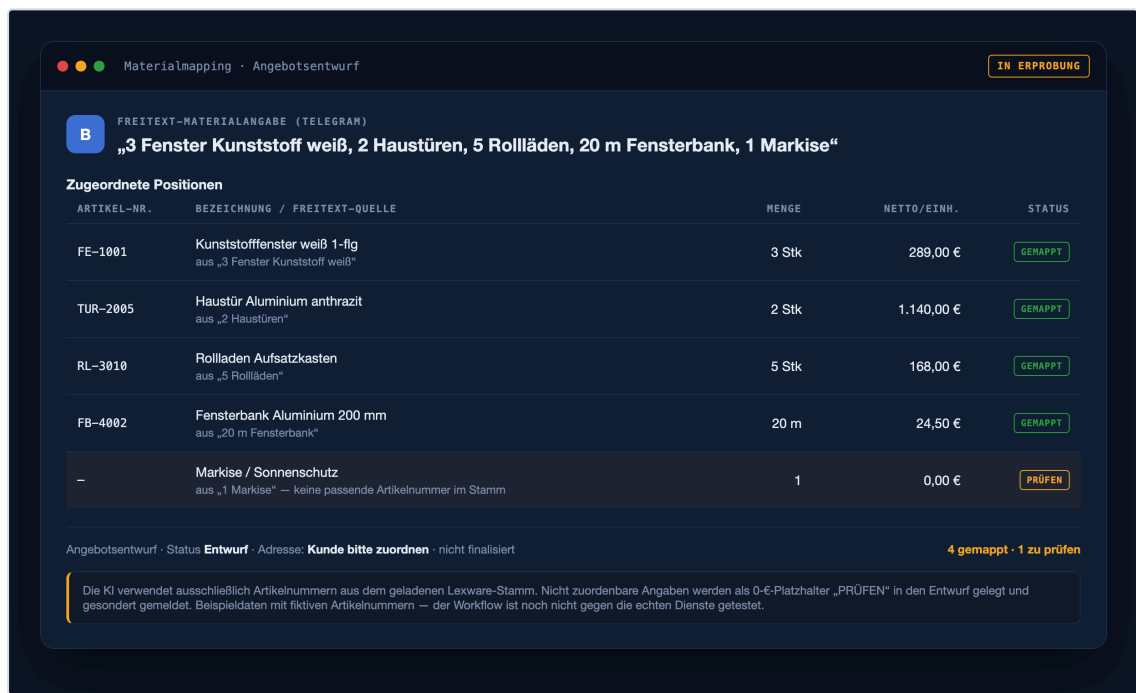


Abbildung 1: Mapping-Ergebnis — vier Positionen gekappt, eine als „PRÜFEN“ markiert. Beispieldaten mit fiktiven Artikelnummern.

Administration

Alle Einstellungen liegen im Node **Config**.

Feld	Bedeutung	Standard
<code>aiBaseUrl</code>	OpenAI-kompatibler Endpunkt, ohne <code>/chat/completions</code>	<code>https://api.openai.com/v1</code>
<code>aiModel</code>	Modell für das Mapping	<code>gpt-4o-mini</code>
<code>seitengroesse</code>	Seitengröße für <code>GET /articles</code>	<code>50</code>
<code>chefEmail</code> / <code>absenderEmail</code>	Empfänger / Absender der E-Mail-Ausgabe	<code>REPLACE_ME@...</code>
<code>telegramChatId</code>	Telegram-Ziel (leer = übersprungen)	<code>REPLACE_ME</code>
<code>whatsappZiel</code> / <code>threemaZiel</code>	WhatsApp-Nummer / Threema-ID (leer = übersprungen)	leer

Zugangsdaten

Nur Platzhalter — echte Werte liegen ausschließlich im n8n-Credential-Store.

Dienst	Credential
Lexware Office	über den API-Baustein (nicht im Workflow)
KI-Endpunkt	Header Auth · <code>Authorization: Bearer <Key></code>
Telegram / WhatsApp / Threema	Eingangs-Trigger + Versand über Messenger-Baustein
E-Mail	SMTP-Credential (Node „Mail: Ausgabe“)

Datenhoheit: Über `aiBaseUrl` kann statt eines Cloud-Anbieters ein lokal betriebenes, OpenAI-kompatibles Modell angesprochen werden.

Datenschutz und Sicherheit

- Verarbeitet werden die **Materialangabe** und eine reduzierte Artikelliste (Nummer, Titel, Einheit, Netto-Preis). Keine Kundendaten im Mapping.
- Empfänger: die eigene n8n-Instanz, die Lexware Office API (über den Baustein), der gewählte KI-Endpunkt, die Messenger-Dienste (Telegram/WhatsApp/Threema) und der SMTP-Server.
- Für maximale Datenhoheit ist ein lokal betriebenes, OpenAI-kompatibles Modell möglich.
- Die KI-Antwort wird gegen den echten Katalog geprüft — **erfundene Nummern werden nie übernommen**.
- Zugangsdaten liegen ausschließlich im n8n-Credential-Store.

Fehlerbehebung

Problem	Mögliche Ursache	Lösung
Fehler beim Baustein-Aufruf	Baustein-Node nicht neu ausgewählt	in den Lexware-Baustein-Nodes und im Node „Baustein: Nachricht senden“ den Workflow im Dropdown neu wählen
KI antwortet nicht / 401	KI-Credential fehlt oder falsch	im Node „KI: Mapping“ Header-Auth-Credential setzen (<code>Bearer <Key></code>)
KI-Antwort nicht als JSON lesbar	Modell liefert Code-Fences/Fließtext	<code>Mapping parsen</code> toleriert Fences; ggf. <code>aiModel</code> wechseln, <code>temperature: 0</code> prüfen
Alle Positionen „PRÜFEN“	Artikelstamm leer / nicht geladen	<code>seitengroesse</code> und Baustein-Antwort prüfen (<code>Bestand geladen?</code>)
Meldung fehlt auf einem Kanal	Ziel in Config leer oder Credential fehlt	Ziel in Config füllen; Messenger-Baustein bzw. SMTP prüfen — leere Ziele werden bewusst übersprungen

Häufige Fragen & Einschränkungen

Über welche Kanäle kann ich schicken?

Telegram, WhatsApp oder Threema — alle als formloser Freitext.

Kann die KI Artikel erfinden?

Nein. Es werden nur Nummern aus dem geladenen Stamm übernommen; Unbekanntes wird „PRÜFEN“.

Muss ich alle Ausgabe-Kanäle nutzen?

Nein. Leere Ziele werden übersprungen; ein einziger gesetzter Kanal (E-Mail oder Messenger) genügt. Das Angebot bleibt Entwurf; Kunde vor Versand zuweisen.

Bekannte Einschränkungen

- Steuersatz fest auf 19 % (die kompakte Liste transportiert keinen Steuersatz).
- Positionen als `type: custom` mit Katalog-Netto-Preis statt echter Artikel-Referenz.
- Platzhalter-Adresse „Materialmapping-Entwurf – Kunde bitte zuordnen“.

Support und Versionshistorie

FlowTrix · Kontakt: <https://flowtrix.de/kontakt/> · Projektübersicht: <https://flowtrix.de/projekte/>

Version	Datum	Änderung
1.1	16.08.2026	Mehrkanal-Version: Eingang über drei Messenger (Telegram, WhatsApp, Threema), Ausgabe über E-Mail + dieselben drei Messenger via Messenger-Baustein. Einsatzbereit.
1.0	22.07.2026	Erstausgabe. Freitext-Empfang, Artikelstamm-Laden, KI-Mapping, Katalog-Prüfung mit PRÜFEN-Markierung, Angebotsentwurf.