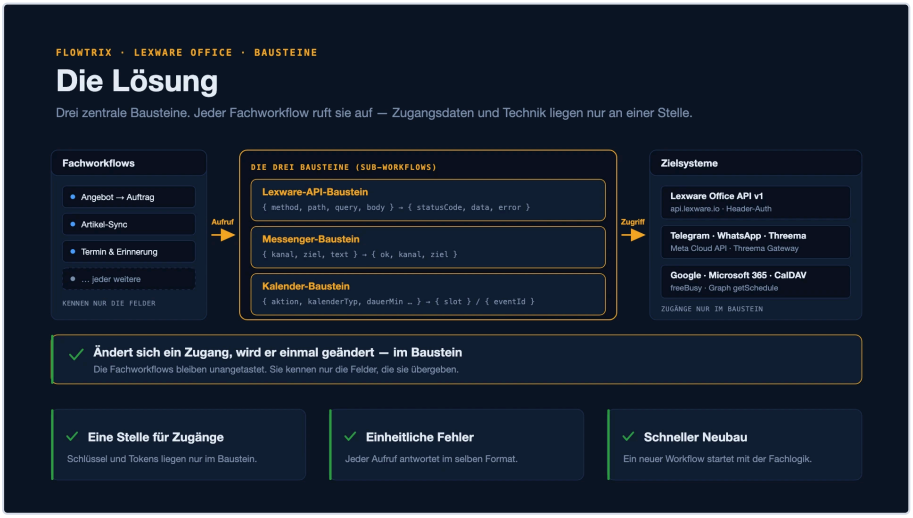
FLOWTRIX · BENUTZER- UND ADMINISTRATORHANDBUCH

Die FlowTrix-Bausteine für Lexware-Workflows

Drei wiederverwendbare n8n-Sub-Workflows für API-Zugriff, Nachrichtenversand und Termine. Einmal einrichten, überall aufrufen.

Version	1.0
Stand	16.08.2026
System	n8n-Sub-Workflows · Lexware Office Public API v1
Herausgeber	FlowTrix · flowtrix.de

Inhaltsverzeichnis



Grundprinzip: Fachworkflows rufen die Bausteine auf — die Zugänge liegen nur dort. Schematische Darstellung.

KAPITEL 1

Was sind die Bausteine?

Ein Baustein ist ein eigener n8n-Workflow, der nur eine Aufgabe hat: die Verbindung zu einem System herzustellen. Er wird nicht direkt gestartet, sondern von anderen Workflows aufgerufen. Der aufrufende Workflow übergibt ein paar Felder und bekommt ein festes Ergebnis zurück.

Grundregeln, die immer gelten:

- Zugangsdaten liegen **nur im Baustein**, nie im Fachworkflow.
- Der Fachworkflow kennt **nur die Felder**, die er übergibt — keine URL, keine Wartezeiten.
- Ändert sich ein Zugang, wird er **einmal** im Baustein geändert.
- Die Antwort hat **immer dieselbe Form** — auch im Fehlerfall.

Es gibt drei Bausteine:

Workflow	Aufgabe
Lexware – API-Call (Baustein)	Zugriff auf die Lexware Office API: Base-URL, Anmeldung, Rate-Limit, Wiederholung bei HTTP 429
Messenger – Nachricht senden (Baustein)	Versand einer Textnachricht über Telegram, WhatsApp oder Threema
Kalender – Verfügbarkeit & Termin (Baustein)	Freien Termin-Slot über eine Kalender-Gruppe suchen oder einen Termin anlegen

KAPITEL 2

Voraussetzungen

- Laufende **n8n-Instanz** (selbst gehostet oder Cloud)
- Für den API-Baustein: **Lexware Office** mit API-Zugang (API-Schlüssel)
- Für den Messenger-Baustein: mindestens einer der Zugänge **Telegram-Bot**, **WhatsApp Business über die Meta Cloud API** oder **Threema Gateway**
- Für den Kalender-Baustein: **Google Calendar** (OAuth2) oder **Microsoft 365** (OAuth2, mit Azure-App-Registrierung)

Hinweis: Es müssen nicht alle Bausteine eingerichtet werden. Wer nur Lexware automatisiert, importiert allein den API-Baustein. Die Bausteine sind voneinander unabhängig.

KAPITEL 3

Erste Schritte

1. Die benötigten Baustein-Dateien in n8n importieren — **immer zuerst den Baustein**, dann die Fachworkflows.
2. Im Node „Lexware Call“ das Credential anlegen: **Header Auth** mit `Authorization` = `Bearer <IHR-API-SCHLÜSSEL>`.
3. Verbindungstest ausführen: der mitgelieferte Workflow „Lexware – Test: Profil abrufen“ ruft einmalig `GET /profile` über den Baustein auf. Ein `statusCode` von `200` und gefüllte `data` bestätigen die Verbindung.
4. Bei Bedarf Messenger- und Kalender-Baustein importieren und deren Zugänge hinterlegen (Kapitel 6 und 7).
5. In den Fachworkflows jeden „Baustein:“-Node öffnen und den Ziel-Workflow im Dropdown neu auswählen.

> **Wichtig nach jedem Import:** In den Execute-Workflow-Nodes steht nach dem Import ein Platzhalter statt einer gültigen Workflow-Kennung. Den Ziel-Workflow im Dropdown einmal neu auswählen — erst dann zieht n8n die in Ihrer Instanz vergebene Kennung.

KAPITEL 4

Rollen und Zuständigkeiten

Rolle	Darf	Braucht
Fachworkflow-Ersteller	Neue Workflows bauen und die Bausteine aufrufen	n8n-Zugang — **keine** API-Schlüssel oder Tokens
Administrator	Bausteine importieren, Credentials pflegen, Konfiguration ändern	n8n-Zugang und die Zugangsdaten der angebundenen Dienste

> Genau darin liegt der Sicherheitsgewinn: Wer einen Workflow baut, braucht die Geheimnisse nicht mehr zu kennen. Er ruft den Baustein auf, und dieser bringt die Anmeldung mit.

KAPITEL 5

Der Lexware-API-Baustein

Der Baustein besteht aus zehn Nodes und führt genau einen API-Aufruf aus — inklusive aller Vorsichtsmaßnahmen, die die Lexware-API verlangt.

5.1 Ein- und Ausgabe

Feld	Bedeutung
`method`	HTTP-Methode, etwa `GET` oder `POST`. Ohne Angabe wird `GET` verwendet.
`path`	Endpunkt hinter der Base-URL, etwa `/quotations`. Die Version `/v1` steckt bereits in der Base-URL.
`query`	Objekt mit Query-Parametern. Leere Werte werden verworfen, der Rest wird kodiert angehängt.
`body`	Objekt für den JSON-Body. Bei lesenden Aufrufen leer lassen.
`statusCode`	HTTP-Status der Antwort.
`data`	Geparster JSON-Body der Antwort.
`error`	Fehlertext im Klartext; leer, solange der Status unter 400 liegt.
`retriesUsed`	Anzahl der automatischen Wiederholungen nach HTTP 429.

5.2 Zentrale Konfiguration

Alle Stellschrauben liegen im Node „Config“:

Einstellung	Wert	Bedeutung
`baseUrl`	api.lexware.io/v1	Basis-Adresse inklusive Version
`rateLimitMs`	550	Mindestabstand zwischen zwei Aufrufen. Rechnerisch genügen 500 ms für zwei Anfragen pro Sekunde; 550 ms sind ein Sicherheitspuffer.
`maxRetries429`	5	Obergrenze für Wiederholungen — verhindert eine Endlosschleife bei dauerhaftem 429
`backoffMs`	1200	Wartezeit vor jedem Wiederholungsversuch

5.3 Was der Baustein automatisch erledigt

- **Vollständige URL bauen:** Base-URL, Pfad und kodierter Query-String werden zusammengesetzt.
- **Wartezeit einhalten:** vor *jedem* Aufruf wird gewartet — auch wenn mehrere Workflows kurz hintereinander zugreifen.
- **Nicht hart abbrechen:** der HTTP-Node liefert auch bei 4xx und 5xx eine Antwort zurück, statt den Lauf zu beenden.
- **Nur bei 429 wiederholen:** andere Fehler wie 401 oder 404 gehen sofort mit Text im Feld `error` zurück.
- **Nur begrenzt wiederholen:** ist die Obergrenze erreicht, wird das Ergebnis regulär zurückgegeben — der aufrufende Workflow entscheidet, wie er damit umgeht.

KAPITEL 6

Der Messenger-Baustein

Zehn Nodes, eine Aufgabe: eine Textnachricht verschicken — unabhängig davon, über welchen Dienst.

6.1 Ein- und Ausgabe

Feld	Bedeutung
`kanal`	`telegram`, `whatsapp` oder `threema`. Ohne Angabe greift der Standardkanal aus der Konfiguration. Groß- und Kleinschreibung spielt keine Rolle.
`ziel`	Empfänger: Telegram-Chat-ID, Telefonnummer oder Threema-ID.
`text`	Die Nachricht als Klartext.
`ok`, `kanal`, `ziel`	Rückgabe an den aufrufenden Workflow.

6.2 Einrichtung der Kanäle

Kanal	Einmalig zu hinterlegen
Telegram	Telegram-Credential im Node „Telegram senden“. Als `ziel` wird die Chat-ID übergeben.
WhatsApp	WhatsApp-Credential im Node „WhatsApp senden“ sowie die Absender-Nummern-Kennung in der Config. Der Versand läuft über die Meta Cloud API.
Threema	Absender-Gateway-ID in der Config und das Gateway-Secret im Node „Threema senden“. Der Versand läuft über das Threema Gateway.

6.3 Verhalten bei Problemen

- **Kein Empfänger übergeben:** der Baustein meldet das zurück, statt einen leeren Versand zu starten.
- **Unbekannter Kanal:** der Lauf stoppt mit der Meldung, welche Kanäle erlaubt sind — ein Tippfehler bleibt so nicht unbemerkt.
- **Fehler beim Versand:** die drei Versand-Nodes sind so eingestellt, dass der Baustein weiterläuft und ein Ergebnis zurückgibt, statt den aufrufenden Workflow abzuwürgen.

> **Standardkanal:** In der Config ist `telegram` voreingestellt. Wer überwiegend einen anderen Dienst nutzt, ändert diesen einen Wert — die Fachworkflows müssen den Kanal dann gar nicht mehr mitgeben.

Der Aufruf im Fachworkflow ist ein Execute-Workflow-Node mit einem Eingabe-Element aus `kanal`, `ziel` und `text`. Für den Versand mehrerer Nachrichten wird der Node so betrieben, dass er je Element einmal ausgeführt wird.

KAPITEL 7

Der Kalender-Baustein

Achtzehn Nodes, zwei Aufgaben: einen freien Termin finden oder einen Termin eintragen — über verschiedene Kalender-Systeme hinweg.

7.1 Die beiden Aktionen

Aktion	Was passiert	Rückgabe
`freibusy`	Die Frei/Belegt-Zeiten aller angegebenen Kalender werden zusammengelegt; im Arbeitszeitfenster wird der erste Slot gesucht, der lang genug ist.	`freiGefunden`, `slotVon`, `slotBis`, `dauerMin`
`anlegen`	Ein Termin wird mit Titel, Beschreibung sowie Start und Ende eingetragen.	`ok`, `eventId`, `link`

7.2 Eingabefelder

Feld	Bedeutung
`kalenderTyp`	`google`, `outlook` oder `caldav`. Ohne Angabe greift der Standardtyp aus der Config.
`kalenderIds`	Die Kalender-Gruppe als Liste oder kommagetrennt; ohne Angabe der Hauptkalender. Bei Microsoft 365 sind das die E-Mail-Adressen der Personen.
`dauerMin`	Benötigte Dauer in Minuten; ohne Angabe 60.
`fensterVon` / `fensterBis`	Suchzeitraum; ohne Angabe ab dem morgigen Tag für 14 Tage.
`titel` / `beschreibung`	Inhalt des Termins; ohne Angabe lautet der Titel „Termin“.
`terminVon` / `terminBis`	Der konkrete Slot, der eingetragen werden soll.

7.3 Konfiguration und Backends

In der Config stehen Standard-Kalendertyp, Arbeitszeitfenster (voreingestellt 08:00 bis 17:00) und Zeitzone (voreingestellt Europe/Berlin). Die Slot-Suche prüft das Arbeitszeitfenster in Schritten von 15 Minuten.

Backend	Stand
Google Calendar	Verdrahtet: Frei/Belegt-Abfrage und Terminanlage. Benötigt ein Google-Calendar-OAuth2-Credential.
Microsoft 365 / Outlook	Verdrahtet über Microsoft Graph: Terminplan-Abfrage und Terminanlage. Benötigt eine Azure-App-Registrierung und ein Outlook-OAuth2-Credential.
CalDAV / Nextcloud	Als Gerüst angelegt und im Ablauf vorgesehen; die eigentliche CalDAV-Abfrage ist noch zu ergänzen. Bis dahin liefert dieser Zweig keinen Slot und legt keinen Termin an.

Unbekannte Kalendertypen und Aktionen stoppen den Lauf mit einer klaren Meldung.

KAPITEL 8

Einen Fachworkflow anbinden

1. Im Fachworkflow einen Node einfügen, der die Eingabefelder setzt — beim API-Baustein zum Beispiel `method`, `path`, `query` und `body`.
2. Dahinter einen **Execute-Workflow-Node** setzen und im Dropdown den gewünschten Baustein wählen.
3. Das Ergebnis auswerten: beim API-Baustein `statusCode` und `error`, beim Messenger-Baustein `ok`, beim Kalender-Baustein `freiGefunden` beziehungsweise `eventId`.
4. Fertig — im Fachworkflow steht kein einziger Zugang und keine Wartezeit.

> **Beispiel aus der Praxis:** Der Artikel- und Preislisten-Sync greift ausschließlich über den API-Baustein auf Lexware zu. Er kennt weder Base-URL noch Schlüssel und muss sich nicht um das Rate-Limit kümmern.

KAPITEL 9

Administration

9.1 Zugang wechseln

Neuer API-Schlüssel, neuer Bot, anderes Postfach: Das Credential wird im jeweiligen Baustein-Node ausgetauscht. Alle Fachworkflows nutzen den neuen Zugang ab dem nächsten Lauf, ohne dass sie angefasst werden.

9.2 Wartezeiten und Wiederholungen anpassen

Sollte Lexware häufiger mit „zu viele Anfragen“ antworten, lassen sich im Config-Node des API-Bausteins der Mindestabstand und die Backoff-Zeit erhöhen. Die Obergrenze für Wiederholungen bleibt sinnvollerweise gesetzt, damit ein dauerhaft überlasteter Dienst nicht endlos angefragt wird.

9.3 Einen weiteren Kanal oder ein weiteres Kalender-Backend ergänzen

Beide Bausteine sind als Verzweigung aufgebaut: eine Auswahl entscheidet, welcher Zweig läuft. Ein neuer Kanal wird als zusätzlicher Zweig ergänzt und in der Auswahl eingetragen — die Fachworkflows übergeben danach einfach den neuen Wert.

9.4 Nach einem n8n-Update

Nach größeren n8n-Aktualisierungen empfiehlt sich ein kurzer Testlauf: der Verbindungstest für den API-Baustein und je eine Testnachricht beziehungsweise eine Slot-Suche. Die Feldnamen für den JSON-Body des HTTP-Nodes können sich zwischen n8n-Versionen unterscheiden und sind dann im Node einmal nachzustellen.

KAPITEL 10

Datenschutz und Sicherheit

- **Zugangsdaten im Credential-Store:** API-Schlüssel und Tokens werden n8n-Credentials zugewiesen und nicht in die Fachworkflows kopiert.
- **Eine Stelle je Dienst:** das erleichtert den regelmäßigen Wechsel von Schlüsseln und den schnellen Entzug.
- **Sparsame Übergabe:** Fachworkflows übergeben nur die Felder, die für den Aufruf nötig sind.
- **Begrenzte Wiederholungen:** der 429-Retry ist gedeckelt — keine Endlosschleife gegen einen überlasteten Dienst.
- **Sichtbarer Abbruch:** fehlende Empfänger, unbekannte Kanäle und unbekannte Kalendertypen führen zu einer klaren Meldung.
- **Selbst gehostet:** n8n läuft auf eigener Infrastruktur; ausgehende Verbindungen bestehen nur zu den bewusst eingerichteten Diensten.

> **Personenbezug beachten:** Messenger- und Kalenderdaten enthalten regelmäßig personenbezogene Daten — Telefonnummern, Namen, Termine. Die Bausteine speichern selbst nichts; die üblichen Pflichten aus der Verarbeitung dieser Daten bleiben davon unberührt und sind im jeweiligen Fachworkflow zu betrachten.

KAPITEL 11

Fehlerbehebung

Symptom	Ursache und Abhilfe
Der Execute-Workflow-Node findet den Baustein nicht	Nach dem Import steht dort ein Platzhalter. Den Ziel-Workflow im Dropdown einmal neu auswählen.
`statusCode` 401 oder 403	Der Schlüssel fehlt, ist abgelaufen oder falsch hinterlegt. Credential im Node „Lexware Call“ prüfen: Header `Authorization` mit vorangestelltem `Bearer`.
`statusCode` 404	Der Pfad ist falsch. `path` beginnt mit einem Schrägstrich und enthält kein `/v1` — das steckt bereits in der Base-URL.
`retriesUsed` ist dauerhaft hoch	Es wird zu schnell aufgerufen. Mindestabstand und Backoff-Zeit in der Config erhöhen oder die Aufrufe im Fachworkflow entzerren.
Nachricht kommt nicht an, ohne Fehlermeldung	Empfängerkennung prüfen: Telegram-Chat-ID, Telefonnummer im erwarteten Format oder Threema-ID. Der Baustein prüft nur, ob ein Ziel übergeben wurde.
„Unbekannter Kanal“ oder „Unbekannter Kalendertyp“	Tippfehler im übergebenen Wert. Erlaubt sind `telegram`, `whatsapp`, `threema` beziehungsweise `google`, `outlook`, `caldav`.
Slot-Suche findet nie einen Termin	Arbeitszeitfenster, Suchzeitraum und gewünschte Dauer prüfen. Beim CalDAV-Zweig ist die Abfrage noch nicht implementiert.

KAPITEL 12

Häufige Fragen

Muss ich alle drei Bausteine einrichten?
Nein. Sie sind voneinander unabhängig. Importiert und eingerichtet wird nur, was gebraucht wird.

Kann ein Baustein direkt gestartet werden?
Er ist zum Aufruf aus anderen Workflows gedacht. Zum Testen gibt es für den API-Baustein den Verbindungstest, der ihn genauso aufruft wie ein Fachworkflow.

Was passiert bei einem Fehler der Lexware-API?
Der Baustein gibt `statusCode` und einen Fehlertext im Feld `error` zurück. Der Fachworkflow entscheidet, ob er abbricht, überspringt oder meldet.

Warum 550 Millisekunden und nicht 500?
Rechnerisch genügen 500 ms für zwei Anfragen pro Sekunde. Die zusätzlichen 50 ms sind ein Sicherheitspuffer gegen Laufzeitschwankungen.

KAPITEL 13

Bekannte Einschränkungen

- Der CalDAV-Zweig des Kalender-Bausteins ist ein Gerüst: Frei/Belegt-Abfrage und Terminanlage sind dort noch zu ergänzen.
- Microsoft 365 setzt eine Azure-App-Registrierung voraus; ohne diese lässt sich das Credential nicht anlegen.
- Messenger- und Kalender-Baustein sind vollständig verdrahtet, aber noch nicht gegen die jeweiligen Dienste durchgetestet — der erste Lauf sollte mit einem Testempfänger beziehungsweise Testkalender erfolgen.
- Der Messenger-Baustein versendet Text. Anhänge und Bilder sind nicht vorgesehen.
- Der API-Baustein führt je Aufruf genau eine Anfrage aus; das Blättern durch lange Ergebnislisten übernimmt der Fachworkflow.

KAPITEL 14

Support und Kontakt

FlowTrix · Digitalisierung, Prozessautomatisierung, IT-Beratung und individuelle Anwendungen
Kontakt und Anfragen über **flowtrix.de/kontakt**

KAPITEL 15

Versionshistorie

Version	Stand	Änderung
1.0	16.08.2026	Erste Fassung: API-, Messenger- und Kalender-Baustein dokumentiert