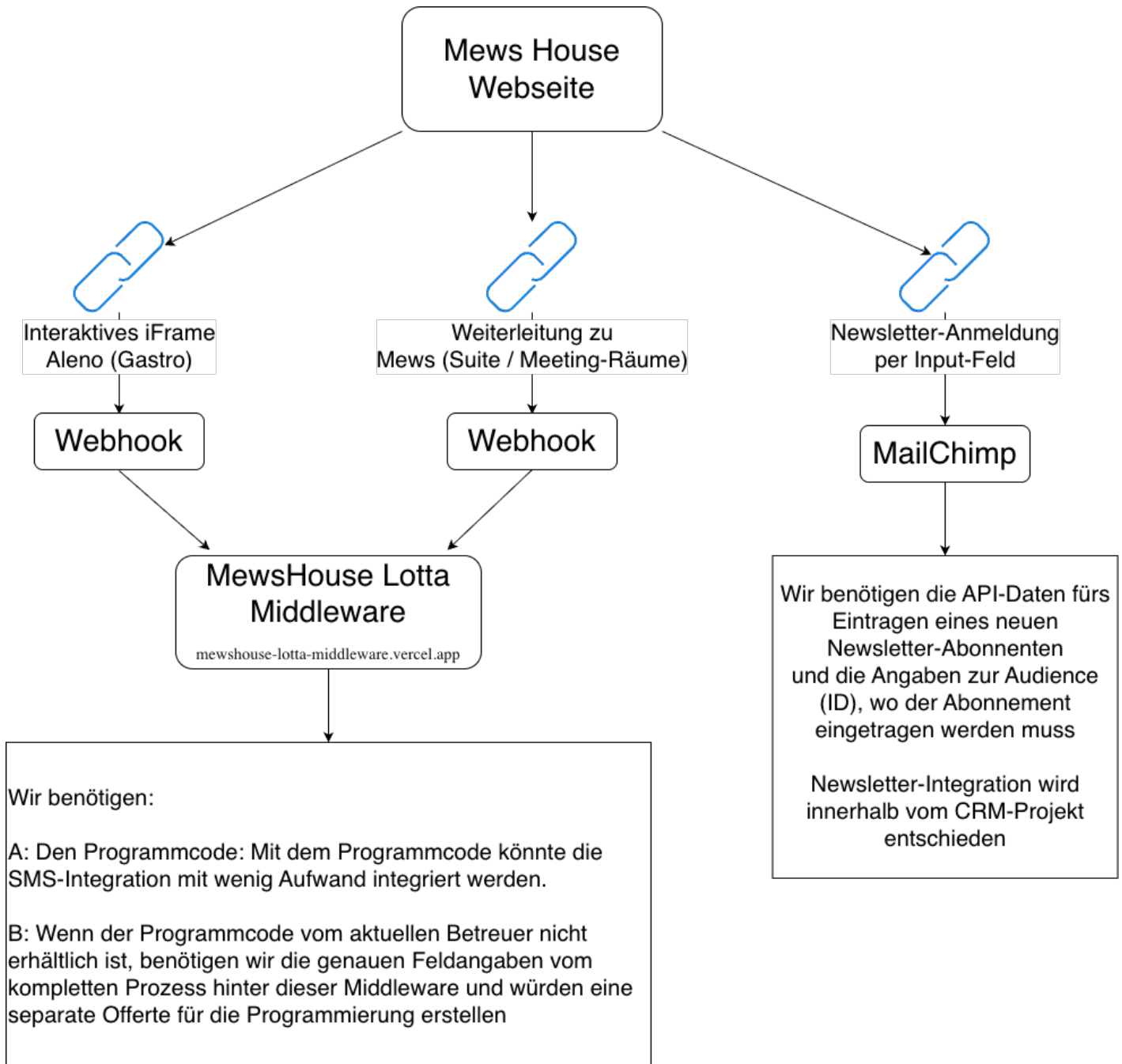


Mews House

- [Webseite](#)
- [Pressmatix Aboverwaltung Integrationsmöglichkeiten](#)
- [Schnittstelle - Bexio](#)
- [Schnittstelle E-Guma](#)
- [Mewshouse Middleware](#)

Webseite

Übersicht



Kontaktformulare

Wir benötigen die E-Mail-Adressen welche die Anfragen von den folgenden Formularen erhalten sollen

- Individuelle Anfragen: <https://mewshouse.go-solution.ch/individuelle-anfragen/>
- Catering: <https://mewshouse.go-solution.ch/anfrage-catering/>
- Event: <https://mewshouse.go-solution.ch/anfrage-event/>
- Blue Bar: <https://mewshouse.go-solution.ch/anfrage-blaue-bar/>
- Mews Table: <https://mewshouse.go-solution.ch/anfrage-mews-table/>

Diverses

- AGB's sind auf der aktuellen Seite nur in Deutsch vorhanden

Newsletter-Integration wird innerhalb vom CRM-Projekt entschieden

Infos zur Lotta Systemintegration:

Guten Tag Herr Novotny

Gerne erläutere ich hier das vorgeschlagene Vorgehen für die Nutzung der API im Objekt [Tiefenhöfe](#) in Kombination mit dem **Lotta Zutrittssystem** und der **Mews PMS-Software**.

Vorschlag: Variante B - Minimal notwendige Rechte für den API-User

Ich empfehle **Variante B**, da diese Lösung die geringsten Systemrechte für den API-User erfordert und gleichzeitig eine saubere, stabile Middleware-Funktion ermöglicht.

Der API-User wird einer **Systemgruppe** zugeordnet, in der ausschliesslich jene Zimmer enthalten sind, welche über Mews reservierbar sind.

Ablauf bei neuen Reservierungen

1. Neuer Gast in Mews

Wird im Mews ein neuer Gast erfasst, z. B. eine Suite für 10 Tage, löst die Middleware beim Antritt automatisch den API-Befehl aus.

2. Visitor-Erstellung in Lotta

Es wird ein *Visitor* angelegt mit:

- **user_system_ids** = jene IDs der Suite (z. B. 1)

3. Zusätzliche Reservationen während des Aufenthalts

Reserviert der Gast beispielsweise einen zusätzlichen Sitzungsraum, erfolgt ein **updateVisitor**, bei dem einfach weitere *user_system_ids* ergänzt werden, z. B. 1, 3.

4. Beendigung temporärer Räume

Sobald der Sitzungsraum nicht mehr benötigt wird, erfolgt erneut ein **updateVisitor**, bei dem die *user_system_id* 3 wieder entfernt wird.

Logik für Start- und Enddaten

- Bei jeder Änderung wird geprüft:
 - **frühestes Startdatum**
 - **spätestes Enddatum**
 - Diese Daten bleiben für den gesamten Zutrittszeitraum massgebend. Hier gilt die gesamte Zeitspanne, die der Gast im Haus mindestens einen Zutritt hat.
 - Die zeitliche Steuerung erfolgt ausschliesslich über **Mews**.
-

API-Befehle in der Lotta-Schnittstelle

Die Middleware führt in Lotta nur die folgenden Aktionen aus:

- **createVisitor**
- **updateVisitor**
- **deleteVisitor**
- **sendAccessCode**

Es werden keine komplexen zeitlichen Berechnungen in der Lotta-API vorgenommen – sämtliche Zeitlogik übernimmt Mews.

Sobald ein Visitor seine Endzeit erreicht hat, wird er gelöscht

Pressmatix Aboverwaltung

Integrationsmöglichkeiten

Zielarchitektur

1. Benutzerauthentifizierung läuft über einen OIDC-Provider (PX-User, Auth0 oder Keycloak)
2. Benutzer Authentifiziert sich in der PressMatrix App oder im Web am Pressmatrix-Kontext [läuft über den OIDC-Provider]
3. Es wird ein Access Token-Generiert
4. Pressmatrix ruft beim Zugriff die eigene Middleware auf
5. Middleware erhält den OIDC-Token

Beispiel-Aufruf durch Pressmatrix

```
curl -X POST https://{auth-server-url}/pmx-api/v2/{profile_token}/authorize \
-H "Authorization: Bearer {ACCESS_TOKEN}" \
-H "OIDC-ID-Token: {OIDC_ID_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{
  "issue_name": "Sample Issue May 2014",
  "issue_date": "2014-05-01",
  "category_name": "Sample Category",
  "category_ids": "20924,20925",
  "product_id_apple": "sample_issue_2014_05",
  "product_id_google": "sample_issue_2014_05",
  "product_id_amazon": "sample_issue_2014_05",
  "product_id_external": "sample_id_1,sample_id_2"
}'
```

Die Middleware prüft zuerst ob der Token gültig ist [Anfrage zum OIDC-Provider]

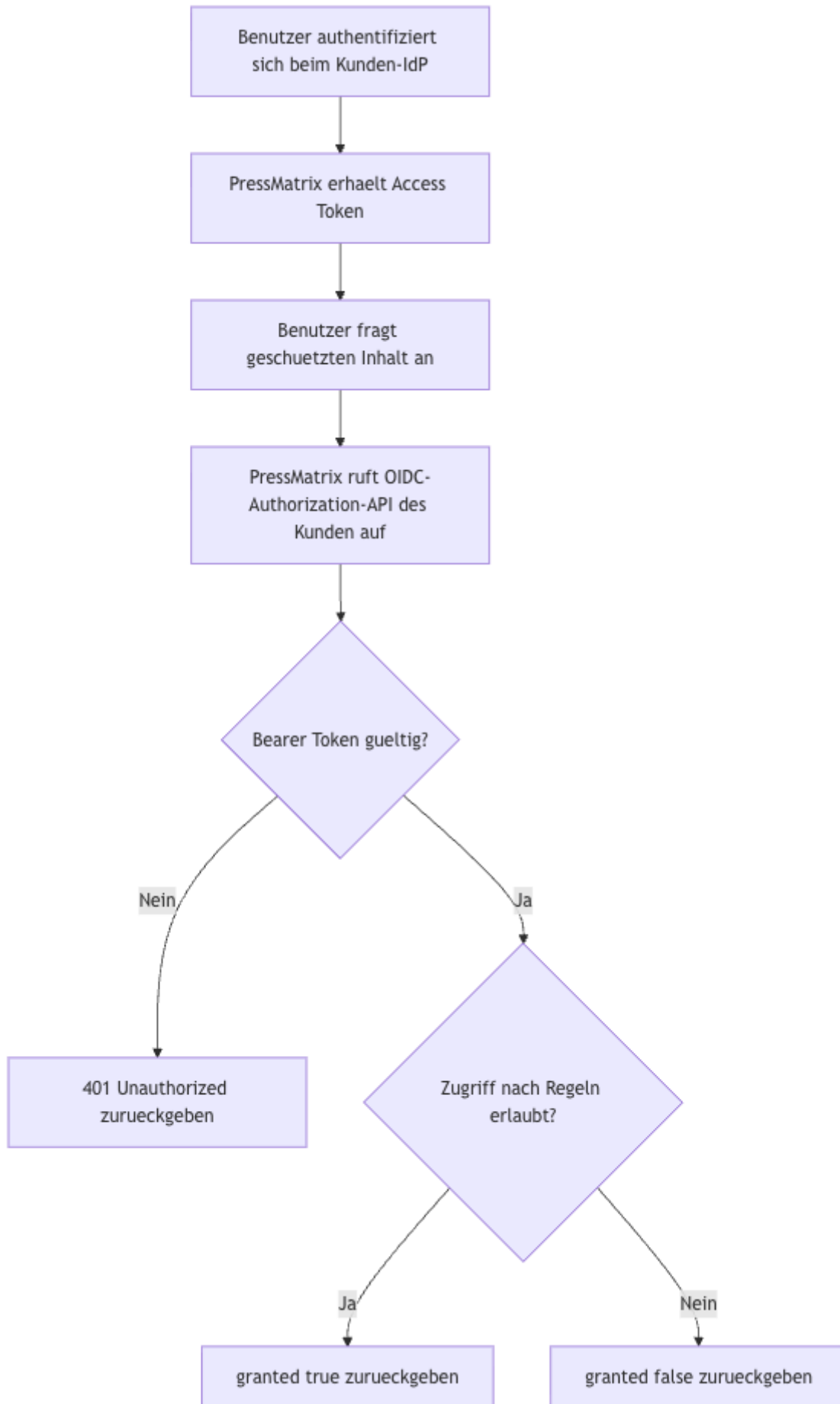
Wenn der Token ungültig ist, wird der Status 401, Unauthorized zurückgegeben

Wenn der Token gültig ist, wird geprüft, ob der Kunde Zugriff auf die angefragte Ressource hat

falls ja: Rückgabe von granted: true

falls nein: Rückgabe von granted: false

Die Daten der Abonnemente müssen in eine eigene Datenbank der Middleware synchronisiert werden, damit beim zugriff nicht auf die Live-Daten im CRM zugegriffen werden müssen. Sonst kann der Kunde bei beispielsweise eines API-Problems bei Sage nicht auf eine Ausgabe zugreifen.



Mögliche CRM Datenstruktur

Kontakt beziehungsweise Benutzer

Feld	Beispiel
CRM Account ID	4711
OIDC Subject	8b153fb2-...
E-Mail	kunde@example.ch
Status	aktiv
Identity Provider	keycloak

Der wichtigste Schlüssel ist der **OIDC-Claim** `sub`. Die E-Mail-Adresse sollte nicht als dauerhafter Primärschlüssel verwendet werden, weil sie sich ändern kann.

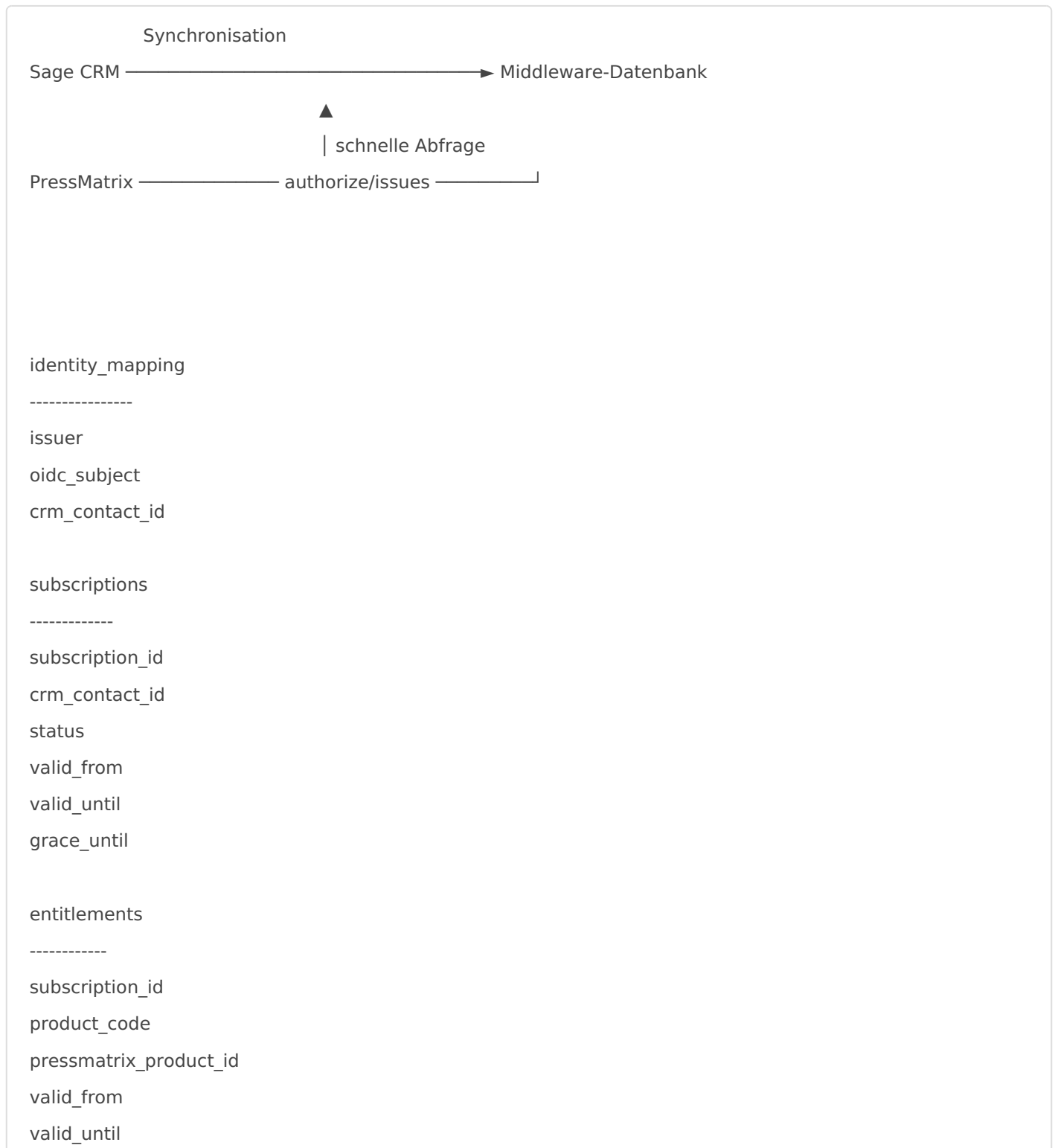
Abonnement

Feld	Beispiel
Subscription ID	SUB-2026-00123
Contact ID	4711
Produktcode	MAGAZIN-PREMIUM
Beginn	2026-01-01
Ende	2026-12-31
Status	aktiv
Kündigungsdatum	optional
Grace Period bis	optional

Produktzuordnung

CRM-Produktcode	PressMatrix-ID
MAGAZIN-PREMIUM	com.publisher.magazin.premium
MAGAZIN-2026	magazin_issue_2026_*
ARCHIV-ZUGANG	Kategorie archive

Mögliche Middleware Datenbank:



Schnittstelle - Bexio

Zeitschätzung vollumfänglich

Bei Komplettintegration in unseren "Sales Data Hub"

Arbeitspaket	Aufwand	Preis
Technische Analyse bexio API und API-Prototyp	16h	
OAuth-/Token-Handling bexio	6h	
Directus-/MySQL-Tabellen und Migrationen	8h	
Generischer bexio API Client in PHP	16h	
Polling, Cursor, Pagination und Retry	16h	
Lookup- und Benutzer-Mappings	??? Wird im CRM über Subtabellen geregelt	
Kontakte Firma ↔ Sage Accounts	20h	
Kontakte Person ↔ Sage Contacts	20h	
Firmen-/Personenbeziehungen	12h	
Dubletten- und Konfliktlogik Kontakte	??? Regeln...	
Auftragskopf bidirektional	20h	
Auftragspositionen bidirektional	25h	
Status-, Steuer- und Rundungsmapping Aufträge	???? müssen wir hier etwas beachten	
Rechnungen → Sage Sales	16h	

Arbeitspaket	Aufwand	Preis
Lösch-/Stornoverhalten	??? allenfalls über Status	
Logging, Monitoring und Fehlerbehandlung	12h	
Integrations- und Regressionstests	8h	
Dokumentation und Deployment	8h	
Gesamtentwicklung exkl. ?	203h à CHF 125.00	CHF 25'375

Grundsätzliche Routine (Accounts, Contacts)

bexio	Sage Sales Management
Kontakt vom Typ Firma	Account
Kontakt vom Typ Person ohne Firmenbezug	Account mit Flag "natürliche Person"
Kontakt vom Typ Person mit Firmenbezug	Contact mit Verknüpfung zu Account
Kunde und Lieferant	Account/Contact mit mehreren Rollen

Verarbeitungsregeln

Regel	Sage Sales Management
bexio contact_type = Firma	Account
bexio contact_type = Person + Firmenrelation vorhanden	Sage Contact unter bestehendem Sage Account
bexio contact_type = Person + keine Firmenrelation	Account mit Flag "natürlicher Person"

Feldzuordnung Bexio "Kontakte" ↔ Sage "Accounts"

bexio Kontakt vom Typ Firma ↔ Sage Account

bexio Kontakt Feld	Sage Account Feld	Richtung	Bemerkung
id	Custom Field <code>bexio_contact_id</code>	B → S	Unveränderbare technische Referenz
Sage Account-ID	Mappingtabelle <code>sage_entity_id</code>	S → Middleware	Nicht zwingend als bexio-Feld verfügbar
nr	<code>accountCode</code> / Custom Field <code>bexio_contact_nr</code>	B ↔ S	Geschäftsnummer, nicht als Primärschlüssel verwenden
contact_type_id	Custom Field <code>contact_type</code>	B → S	Firma/Person
name_1	<code>name</code>	B ↔ S	Firmenname
name_2	<code>additionalName</code> / Custom Field	B ↔ S	Zusatzname oder Abteilung
address	<code>address</code> / <code>address1</code>	B ↔ S	Strasse und Hausnummer
postcode	<code>postalCode</code>	B ↔ S	
city	<code>city</code>	B ↔ S	
country_id	<code>country</code> / <code>countryId</code>	B ↔ S	Lookup-Mapping erforderlich
mail	<code>email</code>	B ↔ S	Firmen-E-Mail
mail_second	Custom Field <code>email_secondary</code>	B ↔ S	Optional
phone_fixed	<code>phone</code>	B ↔ S	
phone_fixed_second	Custom Field <code>phone_secondary</code>	B ↔ S	Optional
phone_mobile	<code>mobilePhone</code>	B ↔ S	Bei Firma eventuell leer
fax	<code>fax</code> / Custom Field	B ↔ S	Falls in Sage vorhanden
url	<code>website</code>	B ↔ S	
remarks	<code>comments</code> / <code>description</code>	B ↔ S	Konflikthanfällig bei bidirektionalem Mapping
language_id	<code>language</code> / Custom Field	B ↔ S	Lookup-Tabelle

bexio Kontakt Feld	Sage Account Feld	Richtung	Bemerkung
<code>user_id</code>	<code>ownerId</code>	B ↔ S	Benutzerzuordnung erforderlich
<code>owner_id</code>	<code>ownerId</code>	B ↔ S	Bedeutung in bexio vorgängig prüfen
<code>is_lead</code>	Account-Klassifikation / Leadstatus	B ↔ S	Businessregel definieren
<code>contact_group_ids</code>	Tags / Klassifikation	B ↔ S	Many-to-many
<code>contact_branch_ids</code>	Branche / Segment	B ↔ S	Lookup-Mapping
<code>updated_at</code>	<code>source_updated_at</code>	B → Middleware	Grundlage für Polling
Sage Änderungszeit	<code>sage_updated_at</code>	S → Middleware	Konflikterkennung
technischer Hash	<code>payload_hash</code>	intern	Verhindert unnötige Updates

Zusätzliche Rollenfelder

Sage-Feld	Werte
<code>business_partner_roles</code>	Kunde, Lieferant, Interessent
<code>is_customer</code>	Boolean
<code>is_supplier</code>	Boolean
<code>bexio_contact_type</code>	Firma, Person
<code>bexio_contact_id</code>	Numerische bexio-ID
<code>bexio_contact_nr</code>	Sichtbare bexio-Kontaktnummer

Feldzuordnung Bexio "Kontakte" ↔ Sage "Contacts"

bexio Kontakt vom Typ Person ↔ Sage Contact

bexio Kontakt Feld	Sage Contact Feld	Richtung	Bemerkung
--------------------	-------------------	----------	-----------

id	Custom Field bexio_contact_id	B → S	Technische Referenz
Sage Contact-ID	Mappingtabelle	S → Middleware	
nr	Custom Field bexio_contact_nr	B ↔ S	Keine technische ID
name_1	lastName	B ↔ S	Zuordnung anhand effektiver bexio-Nutzung prüfen
name_2	firstName	B ↔ S	Kann je Datenpflege umgekehrt verwendet sein
salutation_id	salutation	B ↔ S	Lookup erforderlich
salutation_form	Custom Field salutation_form	B ↔ S	Optional
title_id	title	B ↔ S	Lookup
birthday	birthday	B ↔ S	Datenschutz beachten
address	address	B ↔ S	Privat- oder Geschäftsadresse
postcode	postalCode	B ↔ S	
city	city	B ↔ S	
country_id	country	B ↔ S	Lookup
mail	email	B ↔ S	Empfohlenes Dublettenmerkmal
mail_second	Custom Field email_secondary	B ↔ S	
phone_fixed	phone	B ↔ S	
phone_fixed_second	Custom Field	B ↔ S	
phone_mobile	mobilePhone	B ↔ S	
fax	fax / Custom Field	B ↔ S	
skype_name	Custom Field	B ↔ S	Nur falls weiterhin relevant
remarks	comments / description	B ↔ S	Konfliktregel notwendig
language_id	language	B ↔ S	Lookup
Firmenrelation	accountId	B ↔ S	Über Kontaktbeziehung bestimmen
user_id / owner_id	ownerId	B ↔ S	Benutzer-Mapping
contact_group_ids	Tags / Kategorien	B ↔ S	
updated_at	Middleware-Syncstatus	B → intern	

Kritischer Punkt: Account-Verknüpfung

Für eine Person mit Firmenbezug muss die Reihenfolge lauten:

1. Firmenkontakt in bexio identifizieren.
2. Zugehörigen Sage Account suchen oder erstellen.
3. Person als Sage Contact erstellen.
4. `accountId` des Contacts auf den Sage Account setzen.
5. Beziehung in einer Mappingtabelle speichern.

Eine Person sollte nicht anhand des Firmennamens allein zugeordnet werden. Primär sollte die bexio-Beziehungs-ID beziehungsweise die externe Kontakt-ID verwendet werden.

Feldzuordnung Bexio "Aufträge" ↔ Sage "Salesorders"

bexio stellt Aufträge als Dokumentressource bereit. Für eine vollständige Synchronisation müssen Dokumentkopf und Positionen getrennt betrachtet werden. Die bexio API bietet Verwaltungsfunktionen für Verkaufsaufträge und weitere Verkaufsdokumente.

Auftragskopf

bexio Auftrag Feld	Sage Salesorder Feld	Richtung	Bemerkung
<code>id</code>	Custom Field <code>bexio_order_id</code>	B → S	Technischer Schlüssel
Sage Salesorder-ID	Mappingtabelle	S → intern	
<code>document_nr</code>	<code>orderNumber</code>	B ↔ S	Sichtbare Auftragsnummer
<code>title</code>	<code>name</code> / <code>title</code>	B ↔ S	
<code>contact_id</code>	<code>accountId</code>	B ↔ S	Über Kontaktmapping
<code>contact_sub_id</code>	<code>contactId</code>	B ↔ S	Ansprechpartner
<code>user_id</code>	<code>ownerId</code>	B ↔ S	Benutzer-Mapping
<code>project_id</code>	<code>projectId</code> / Custom Field	B ↔ S	Optional
<code>language_id</code>	<code>language</code>	B ↔ S	Lookup
<code>currency_id</code>	<code>currency</code>	B ↔ S	Lookup, ISO-Code bevorzugen
<code>payment_type_id</code>	<code>paymentTerms</code>	B ↔ S	Lookup
<code>bank_account_id</code>	Custom Field	B ↔ S	Nur falls fachlich benötigt
Dokumentdatum	<code>orderDate</code>	B ↔ S	Feldname in bexio prüfen
<code>is_valid_from</code>	<code>validFrom</code>	B ↔ S	Optional
<code>is_valid_to</code>	<code>validUntil</code>	B ↔ S	Optional

bexio Auftrag Feld	Sage Salesorder Feld	Richtung	Bemerkung
header	headerText / Beschreibung	B ↔ S	
footer	footerText / Beschreibung	B ↔ S	
contact_address	Rechnungsadresse	B ↔ S	Snapshot des Dokuments
delivery_address	Lieferadresse	B ↔ S	
delivery_address_type	Lieferadresstyp	B ↔ S	Lookup
total_net	netAmount	B → S	Berechnetes Feld möglichst nicht zurückschreiben
total_taxes	taxAmount	B → S	
total_gross	grossAmount	B → S	
mwst_type	taxType	B ↔ S	Lookup
mwst_is_net	pricesIncludeTax	B ↔ S	Boolean-Transformation
show_position_taxes	Custom Field	B ↔ S	Darstellungsinformation
kb_item_status_id	status	B ↔ S	Statustabelle erforderlich
api_reference	externalReference	B ↔ S	Sehr gut für Integrationsreferenz geeignet
updated_at	Syncstatus	B → intern	Polling
Sage Änderungszeit	Syncstatus	S → intern	Konflikterkennung

Auftragspositionen

bexio Position Feld	Sage Salesorder Line Feld	Richtung	Bemerkung
Positions-ID	externalLineId	B → S	Mappingtabelle
amount	quantity	B ↔ S	
unit_id	unit	B ↔ S	Lookup
text	description	B ↔ S	
unit_price	unitPrice	B ↔ S	
discount_in_percent	discountPercent	B ↔ S	
Artikelreferenz	productId / articleId	B ↔ S	Nur mit Artikelmapping
Steuersatz-ID	taxRate / taxId	B ↔ S	Lookup
Positionssumme	lineTotal	B → S	Berechnetes Feld
Sortierung	position / sortOrder	B ↔ S	

bexio Position Feld	Sage Salesorder Line Feld	Richtung	Bemerkung
Positionstyp	lineType	B ↔ S	Normal, Titel, Zwischensumme, Text etc.

Feldzuordnung Bexio Rechnungen ↔Sage "Sales"

Hier würde es Sinn machen nur die Kopfdaten zu übermitteln

bexio Rechnung Feld	Sage Sales Feld	Richtung	Bemerkung
id	Custom Field <code>bexio_invoice_id</code>	B → S	Technische Referenz
document_nr	<code>saleNumber</code> / <code>invoiceNumber</code>	B → S	
title	<code>name</code> / <code>title</code>	B → S	
contact_id	<code>accountId</code>	B → S	Über Mapping
contact_sub_id	<code>contactId</code>	B → S	Ansprechpartner
user_id	<code>ownerId</code>	B → S	Benutzer-Mapping
project_id	Custom Field / Projekt	B → S	Optional
currency_id	<code>currency</code>	B → S	Lookup
language_id	Custom Field	B → S	
Rechnungsdatum	<code>saleDate</code> / <code>invoiceDate</code>	B → S	
Fälligkeitsdatum	<code>dueDate</code>	B → S	
payment_type_id	<code>paymentTerms</code>	B → S	Lookup
header	<code>description</code> / Custom Field	B → S	
footer	Custom Field	B → S	
total_net	<code>netAmount</code>	B → S	

bexio Rechnung Feld	Sage Sales Feld	Richtung	Bemerkung
total_taxes	taxAmount	B → S	
total_gross	grossAmount	B → S	
erhaltene Zahlungen	paidAmount	B → S	Falls Sales dies unterstützt
offener Betrag	openAmount	B → S	
Rechnungsstatus	status	B → S	Entwurf, offen, bezahlt, storniert
mwst_type	taxType	B → S	
mwst_is_net	pricesIncludeTax	B → S	
contact_address	Rechnungsadresse	B → S	Dokument-Snapshot
api_reference	externalReference	B → S	
updated_at	Syncstatus	B → intern	

Polling-Intervalle für alle Entitäten

Entität	Intervall
Kontakte	alle 5–15 Minuten
Aufträge	alle 5 Minuten
Rechnungen	alle 5–15 Minuten
Lookup-Daten	täglich
vollständiger Abgleich	nachts oder wöchentlich

Schnittstelle E-Guma

Laut Schnittstellendokumentation können Kunden oder Firmen nicht explizit abgeholt werden. Kundendaten sind immer Bestandteil einer Bestellung..

Die Schnittstellen-Integration ist nach diesem Kriterium "fragwürdig"

- Gutscheinbestellungen können zeitlich gefiltert abgerufen werden.
- Einzelne Bestellungen können abgerufen werden.
- Neue Gutscheinbestellungen können angelegt werden.
- Für neu erstellte Bestellungen existiert ein Webhook.
- Kundendaten sind Bestandteil einer Bestellung.
- Es sind **keine öffentlich dokumentierten, eigenständigen CRUD-Endpunkte für Kunden oder Firmen** ersichtlich.

Realisierbarkeit

Gewünschte Entität	Realisierbarkeit	Empfohlene Richtung
Accounts	Teilweise, nur über Bestellung nicht sinnvoll weil keine eindeutige Kennung vorhanden ist	e-guma → Sage Sales Management
Kontakte	Teilweise, nur über Bestellung nicht sinnvoll weil keine eindeutige Kennung vorhanden ist	e-guma → Sage Sales Management
Sales	Gut möglich	e-guma → Sage Sales Management
Aufträge	Gut möglich	e-guma → Sage Sales Management; optional Sage → e-guma
Gutschein-/Produktstammdaten	Möglich	e-guma → Sage Sales Management
Marketing-Einwilligung	Möglich	e-guma → Sage Sales Management

Die Integration ist unidirektional theoretisch möglich

Accounts und Kontakte

Die e-guma-Bestellungen enthalten ein eingebettetes `customer`-Objekt mit:

- Kunden-ID
- Anrede
- Vorname
- Nachname
- Firma
- Adresse
- E-Mail
- Telefon
- Marketing-Einwilligung
- Zeitpunkt des Double-Opt-ins

Diese Informationen können in Sage Sales Management als Accounts und Contacts erstellt beziehungsweise aktualisiert werden.

Zeitschätzung

Minimaler sinnvoller Umfang

Enthalten:

- e-guma → Sage Accounts
- e-guma → Sage Contacts
- e-guma → Sage Sales
- Polling
- Datenbanktabellen
- Mapping
- Logging
- Basis-Tests

Schätzung: 90-135 Stunden

Empfohlener produktiver Umfang

Zusätzlich:

- Aufträge und Positionen
- Webhook
- Stornierungen
- robuster Delta-Abgleich

- umfassende Testfälle
- Monitoring und bessere Fehlerbehandlung

Schätzung: 130-195 Stunden

Erweiterter Umfang

Zusätzlich:

- Sage → e-guma Gutscheinbestellungen
- Verarbeitung von Gutscheincodes/PDF-URLs
- erweiterte Konfigurationsoberfläche
- Initialimport-Assistent
- manuelles Replay fehlgeschlagener Datensätze

Schätzung: 160-235 Stunden

**Geschätzte Entwicklung
gesamte Anbindung mit allem:
380 - 565 Stunden**

Mewshouse Middleware

Todo:

Tabelle der Gäste inkl. Bearbeitung

Tabelle der Zugänge inkl. Bearbeitung

Logging-Tabelle zur reinen Ansicht

Webhooks hinzufügen

Mews Webhook anpassen: Noch nicht analysiert... (zusätzlicher Webhook, wenn möglich)

aleno prod webhook id: fc1c3d1e90280b42c72f2706 (zusätzlicher Webhook, wenn möglich)

Crons einrichten

Damit der Zugriff funktioniert muss "CRON_SECRET" in den Environment-Variablen im header "authorization" als Bearer mitgegeben werden!

Endpunkt	Beschreibung	Zeiten
/api/lotta/access-groups	Access-Groups synchronisieren	06:00, 12:00, 18:00
/api/aleno/sync-reservations	Aleno-Reservierungen synchronisieren	06:00, 12:00, 18:00
/api/mews/sync-reservations	Mews-Reservierungen synchronisieren	06:00, 12:00, 18:00
/api/guests/process	Gäste-Zugriffe synchronisieren / aktualisieren	alle 5 Minuten

Testszenarien

Mews:

1. Anders als beim Aleno-Flow muss hier das Mews-Dashboard verwendet werden, um eine Testbuchung anzulegen:

<https://app.mews.com/Commander/dbbef1f9-d4c5-4324-be7e-b3680081bef3/Reservation/Create>

Eine neue Buchung erstellen, idealerweise mit einem Datum in der weiter entfernten Zukunft. Als Nachname «Test» verwenden. Unbedingt die eigene Handynummer angeben, damit der Lotta-SMS-Versand funktioniert.

2. Sicherstellen, dass die Buchung den Status «Bestätigt» hat.
3. Lotta-Middleware-Dashboard öffnen:

URL: <https://mewshouse-lotta-middleware.vercel.app/>

Login: Admin / M3W5H0U53L0TT7

→ Guests → entsprechendes Gastprofil suchen (falls noch nicht vorhanden, ein paar Minuten warten) → öffnen

→ Prüfen, ob Start- und Enddatum korrekt sind

→ Unter «Accesses» prüfen, ob ein entsprechender «Mews»-Eintrag vorhanden ist

→ Eintrag öffnen und sicherstellen, dass «Type» mit der gebuchten Suite übereinstimmt

4. (Optional: Sofort-Zugang testen)

- Im Gastprofil den «Type» von «Automated» auf «Manual» ändern
- Start- und Enddatum manuell so anpassen, dass sofortiger Zugang möglich ist (Startdatum in der Vergangenheit, Enddatum in der Zukunft)
- Unter «Access Types» die gebuchte Suite manuell auswählen
- Speichern und ein paar Minuten warten
- Lotta Manager Dashboard öffnen:
<https://manager.lottasystems.app/users/details/599>
→ Benutzer → API Tiefenhöhe → «Access» → «Access Codes»
→ Prüfen, ob ein Eintrag für den Gast vorhanden ist
- Falls der Nachname «Test» verwendet wurde, prüfen, ob eine SMS eingegangen ist (Versand frühestens 24h vor Startdatum)
- Wenn du fertig bist mit Testen, den «Type» wieder auf «Automated» und speichern.

Aleno:

1. <https://www.mewshouse.ch/> → Mews Table buchen

Ebenfalls ein Datum in der Zukunft wählen. Als Nachname «Test» verwenden und die eigene Handynummer angeben. Kreditkartenzahlung starten und anschliessend

abbrechen.

2. Aleno-Dashboard öffnen:

<https://mytools.aleno.me/aleno3/booqIn?allTimeslotsExpanded=yes&statusFilter=no&showTimeslots=yes>

→ booqin → Datum auswählen → STATUS FILTER OFF

→ Buchung suchen und den Status von «CC Requested» auf «Confirmed» ändern → speichern

3. Lotta-Middleware-Dashboard öffnen (siehe oben)

→ Guests → Gastprofil suchen → öffnen

→ Prüfen, ob Start- und Enddatum korrekt sind

→ Unter «Accesses» prüfen, ob ein entsprechender «Aleno»-Eintrag vorhanden ist

→ Sicherstellen, dass «Type» = «Restaurant»

4. (Optional: Sofort-Zugang testen)

- Im Gastprofil den «Type» von «Automated» auf «Manual» ändern
- Start- und Enddatum manuell so anpassen, dass sofortiger Zugang möglich ist
- Unter «Access Types» den passenden Zugang manuell auswählen
- Speichern und ein paar Minuten warten
- Lotta Manager Dashboard öffnen:

<https://manager.lottasystems.app/users/details/599>

→ Benutzer → API Tiefenhöhe → «Access» → «Access Codes»

→ Prüfen, ob ein Eintrag vorhanden ist

- Falls der Nachname «Test» verwendet wurde, prüfen, ob eine SMS eingegangen ist (Versand frühestens 24h vor Startdatum)
- Wenn du fertig bist mit Testen, den «Type» wieder auf «Automated» und speichern.

Im Anschluss die Mews-/Aleno-Buchungen in den jeweiligen Systemen stornieren. Anschliessend ein paar Minuten warten und sicherstellen, dass sich die Einträge im Lotta-Middleware-Dashboard sowie im Lotta ManagerDashboard entsprechend aktualisieren.