

TECHNISCHE DOKUMENTATION

WAMAS Lift und Store zu Business Central

REST und OAuth2
Testnachweise und Anforderungen an den Empfänger

AcuMax Med AG | Testumgebung | Version 1.0 | 23. September 2026

Ergebnis und Handlungsbedarf

Der Zugriff auf die Business Central Custom API wurde in Postman grundsätzlich hergestellt. Der Gesprächsverlauf dokumentiert einen erfolgreichen OAuth2-Tokenabruf, einen erfolgreichen GET sowie einen Minimal-POST mit HTTP 201. Der entscheidende POST mit einem aus dem NAS-Archiv rekonstruierten WAMAS-Payload wurde jedoch mit HTTP 404 und Internal_RecordNotFound abgewiesen. Der Fehler nennt den fehlenden Datensatz ACU WAMAS Inbound Message mit Entry No.=0. [G, B4]

Damit ist die direkte Übernahme vollständiger WAMAS-Nachrichten in Business Central noch nicht abgenommen. Der ERP-Partner muss die Annahme des nativen WAMAS-Formats, die interne Parent-Erzeugung, die Zuordnung der Items und die Antwort an den WAMAS-Sender implementieren beziehungsweise korrigieren. Ein vorgeschalteter Parent-POST durch WAMAS ist im nachgewiesenen Referenzvertrag nicht vorgesehen. [Q1, Q2]

Zweck und Zielgruppe

Diese Übergabe richtet sich an den ERP-/Business-Central-Partner und SSI/WAMAS-Techniker. Sie beschreibt die Testfolge, ordnet die Belege ein und definiert den gewünschten Schnittstellenvertrag sowie prüfbare Abnahmekriterien. Die Produktivfreigabe setzt einen erfolgreichen Test mit dem tatsächlichen WAMAS-Sender voraus.

Leseführung

Abschnitt	Inhalt
1 bis 3	Ausgangsdaten, zeitlicher Testablauf und Testmatrix
4 bis 5	Receiver-Code und echte WAMAS-Beispiele
6 bis 8	Soll-Architektur, BC-Implementierung und Abnahme
9 und Anhang	Quellenverzeichnis und vier Screenshot-Belege

Belegstufen: DIREKT = Screenshot, Datei oder wörtliche Nutzerausgabe geprüft. GESPRÄCH = Ergebnis im bisherigen Gespräch dokumentiert. ANALYSE = technische Schlussfolgerung. SOLL = Implementierungsempfehlung oder noch offener Abnahmepunkt. Die Kennzeichnungen verhindern, dass ein Testbefund mit einer bestätigten Fehlerursache verwechselt wird.

1 Ausgangsdaten und OAuth2

Parameter	Wert im Test
Tenant	ab9cc3ad-19f4-488d-b606-bd5b9bb396fc
Environment	Test
Client ID	5c546755-2749-4ec5-96c7-7ef72aac8b3c
Company Name	AcuMax Med AG
Company ID	9fc70257-de96-eb11-8ce5-000d3a4c16d8
Custom API	Publisher acumax · Gruppe wamas · Version v2.0
EntitySet	wamasInboundMessages
Scope	https://api.businesscentral.dynamics.com/.default
Grant Type	client_credentials
Client Secret und Access Token	Nicht Bestandteil dieser Dokumentation; geschützt konfigurieren.

Die Company-ID stammt aus der vom Nutzer eingefügten companies-Antwort. Die erfolgreiche Custom-API-Variante verwendet den Company-Namen als Query-Parameter. Ein Aufruf dieser Custom API über die Company-GUID ist in der Testfolge nicht nachgewiesen. [G, B2]

Endpunkte

Für gut lesbare Request-Beispiele wird im Folgenden `{{bcBase}}` verwendet:

```
https://api.businesscentral.dynamics.com/v2.0/
ab9cc3ad-19f4-488d-b606-bd5b9bb396fc/Test
```

Die Zeilen oben bilden eine einzige URL ohne Zeilenumbruch. Custom API:

```
{{bcBase}}/api/acumax/wamas/v2.0/wamasInboundMessages
?company=AcuMax%20Med%20AG
GET {{bcBase}}/api/v2.0/companies
GET {{bcBase}}/api/acumax/wamas/v2.0/$metadata
```

Tokenabruf in Postman

```
POST https://login.microsoftonline.com/
      ab9cc3ad-19f4-488d-b606-bd5b9bb396fc/oauth2/v2.0/token
Content-Type: application/x-www-form-urlencoded
```

```
grant_type = client_credentials
client_id  = 5c546755-2749-4ec5-96c7-7ef72aac8b3c
client_secret = <geschützt hinterlegter Secret-Wert>
scope      = https://api.businesscentral.dynamics.com/.default
```

GESPRÄCH: Der Token-Test ergab 200 OK, token_type=Bearer und expires_in=3599. Das beweist die damalige Tokenausstellung, aber allein noch keine BC-Objektberechtigung oder Payload-Verarbeitung. Die Access-Token-Laufzeit ist aus der jeweiligen Antwort zu übernehmen. [G, M1]

OFFEN: Die gelieferte JWK-URL verweist auf Tenant ce896e4e-701d-4691-8bbb-e7059c7dc33d und damit auf einen anderen Tenant als Token-URL und BC-URL. Sie wurde im manuellen Postman-Test nicht verwendet. SSI und ERP-Partner müssen die passende Discovery-/JWK-Konfiguration für die eingesetzte WAMAS-Version vor Inbetriebnahme bestätigen. [G]

2 Chronologischer Testablauf

Die Postman-Testfolge fand am 23.09.2026 statt. Zeitangaben unten bezeichnen die zugehörigen Gesprächsbeiträge in Europe/Zürich (UTC+2), keine separat gemessenen Serverzeitpunkte. Die NAS-Nachrichten wurden bereits im August und September 2026 archiviert. [G, Q2]

Zuerst Authentifizierung und Company

10:16 bis 10:24 – Die vom ERP-Partner gelieferten Verbindungsdaten wurden in Postman eingerichtet. Der separate Client-Credentials-POST an Microsoft Entra lieferte 200 OK mit Bearer Token. [G; GESPRÄCH]

Danach – Die ersten BC-GETs lieferten 401 Unauthorized mit „The credentials provided are incorrect“. Der genaue Auslöser ist nicht abschliessend belegt. Ein fehlender oder nicht übernommener Bearer Token wurde beim neuen companies-Request vermutet; eine tatsächlich vorgenommene Rechteänderung beim ERP-Partner ist nicht dokumentiert. Fehlendes Accept allein ist keine belegte Ursache. [G; GESPRÄCH/ANALYSE]

10:32 bis 10:41 – Nach dem Authentifizierungsschritt erschien 404 Internal_CompanyNotFound. Die companies-Abfrage identifizierte AcuMax Med AG und die Company-ID. Mit company=AcuMax Med AG gelang anschliessend der GET auf wamasInboundMessages. Der Company-Fehler trat später bei einem GET ohne Company erneut auf und ist direkt im Screenshot belegt. [G, B1, B2]

Danach Datenmodell und Schreibtests

10:44 bis 10:48 – Die Custom-API-Metadaten wurden ausgewertet; anschliessend wurden die vorhandenen Parents mit \$expand=items abgefragt. Die Navigation war lesbar, die Items waren leer. [G; GESPRÄCH]

10:54 – Der Minimal-POST mit {"source":"REST"} lieferte 201 Created. Die Antwort enthielt jedoch eine Null-GUID, entryNo=0 und einen leeren status. Deshalb bestätigt diese Antwort allein keine verarbeitete WAMAS-Nachricht und keine korrekte Rückgabe des gespeicherten Schlüssels. [G; GESPRÄCH]

```
{
  "id": "00000000-0000-0000-0000-000000000000",
  "entryNo": 0, "status": "", "source": "REST"
}
```

11:02 – Ein Deep Insert mit source=REST und items[0] scheiterte mit 404 Internal_RecordNotFound, Entry No.=0. data enthielt ein gekürztes WLSSDACK002 zu IR000007; dies war ein Strukturtest und kein vollständiger fachlicher WAMAS-Test. [B3; DIREKT]

11:07 – Ein Kontroll-GET mit Company, \$orderby=entryNo desc, \$top=5 und \$expand=items lieferte 200 OK. Sichtbar waren entryNo 2 und 1, status „Message received“, source „REST“ und jeweils items=[]. Ein zusätzlich sichtbarer Parent aus dem fehlerhaften Deep Insert wurde nicht gefunden. Ein vollständiger Transaktions-Rollback ist daraus allein nicht beweisbar. [B2; DIREKT/ANALYSE]

Zum Schluss Abgleich mit echten WAMAS-Daten

11:16 bis 11:23 – Das NAS-Archiv und anschliessend main.py klärten den äusseren Request. Die Zwischenvermutung, WAMAS sende ein einzelnes Item ohne items-Array, wurde durch den Receiver-Code korrigiert. [G, Q1, Q2]

11:29 – Der erneute POST ohne source, mit WAMAS-ID 505 und dem archivierten WLSSTOCKADJ001 als data-String, ergab erneut 404 Internal_RecordNotFound, Entry No.=0. Damit reproduziert auch der aus echten Daten rekonstruierte native Aufbau denselben Fehler. Ein direkter WAMAS-zu-BC-Lauf wurde damit noch nicht durchgeführt. [B4; DIREKT]

3 Testmatrix und Fehlerbewertung

Test	Request und Prüfung	HTTP	Ergebnis und Beleg
T01	Token-POST mit Client Credentials	200	Token ausgestellt; Bearer, 3599 s. GESPRÄCH [G].
T02	Erste GETs an BC bzw. companies	401	Zugang abgewiesen. Genaue Ursache offen; Tokenübernahme als Hypothese. [G]
T03	GET ohne wirksame Company	404	Internal_CompanyNotFound. Company fehlt oder wird nicht aufgelöst. DIREKT [B1].
T04	GET Standard API companies	Erfolg ¹	Company-Name und GUID in Originaltext nachgewiesen. DIREKT [G].
T05	GET Custom API mit Company	200	Parents lesbar. Erster Erfolg im Gespräch; Kontroll-GET direkt belegt. [G, B2]
T06	GET Custom API \$metadata	XML ¹	Parent und Child-Felder im Gespräch dokumentiert. [G]
T07	GET mit \$expand=items	200	Zwei Parents; beide items leer. DIREKT [B2].
T08	POST {"source":"REST"}	201	Minimal-POST angenommen; Antwortwerte Null-GUID/0/leer. GESPRÄCH [G].
T09	POST source und items mit gekürztem WLSSDACK002	404	Internal_RecordNotFound; Parent Entry No.=0. DIREKT [B3].
T10	Kontroll-GET nach T09	200	Weiterhin entryNo 2 und 1, items leer. DIREKT [B2].
T11	NAS-Code und Archiv auswerten	—	Schema und originale Nutzdaten belegt; kein neuer HTTP-Test. [Q1, Q2]
T12	POST natives items-Format mit Originaldaten von ID 505	404	Gleicher Parent-Fehler auch ohne source. DIREKT [B4].

¹ Erfolgreiche Daten-/Metadatenrückgabe ist dokumentiert; ein separat verifizierbarer Statuscode liegt für diese beiden Schritte nicht vor. Keine 200-Angabe wird ergänzt, nur weil sie zu erwarten wäre.

Drei Fehlerklassen sauber unterscheiden

401 betrifft den Zugang zum Dienst. Internal_CompanyNotFound betrifft die Auswahl der Firma.

Internal_RecordNotFound mit der benannten ACU-Tabelle betrifft dagegen die Datensatzverarbeitung innerhalb der Custom API. Ein 404 bedeutet hier nicht automatisch, dass die URL oder der Server nicht existiert.

ANALYSE: Beide fehlgeschlagenen POSTs lenken die Fehlersuche auf den Lebenszyklus des Parent-Datensatzes und die Child-Zuordnung. Die konkrete AL-Codezeile, Trigger-Reihenfolge und ein möglicher temporärer Datensatz sind ohne BC-Code und Telemetrie nicht bewiesen. Ebenso beweist der frühe Parent-Fehler noch nicht die fachliche Gültigkeit sämtlicher Telegrammfelder.

4 Referenzreceiver und natives Schema

DIREKT: main.py definiert den WAMAS-Eingang als POST /di-api/v3/HOST/messages. Die separate Route /api/v3/messages ist nicht derselbe POST-Vertrag: Dort erwartet der Receiver type und payload; der GET dieser Route liefert das Archiv. [Q1]

Die Modelle definieren den äusseren Request

```
class WamasItem(BaseModel):
    id: int | str
    converter: str
    data: Any
    processingPath: str | None = ""
```

```
class WamasMessageBatch(BaseModel):
    items: list[WamasItem]
```

Der Code belegt das akzeptierte Batch-Schema; das Archiv belegt die gespeicherten echten WAMAS-Inhalte. Zusammen erlauben sie die Rekonstruktion des nativen Request-Aufbaus. Ein unveränderter HTTP-Mitschnitt mit Headern und Bytefolge liegt damit nicht vor. Insbesondere lässt das Python-Modell data allgemein zu; in den ausgewählten echten Nachrichten ist data ein JSON-String. [Q1, Q2]

Feld	Belegte Bedeutung
items	Äussere Liste; ein oder mehrere WamasItem-Objekte.
id	WAMAS-Nachrichten-ID; Python akzeptiert int oder str. Keine BC Entry No.
converter	Im Archiv JsonConverter. Ein BC-Converter muss diesen Wert unterstützen.
data	Vollständiges DI_TELEGRAMS-JSON als String in den Archivbeispielen.
processingPath	Im Archiv leer. Receiver normalisiert None bzw. leer zu "".

Archivierung verändert den Feldnamen

```
store_message("v3", "WAMAS", {
    "wamas_message_id": item.id,
    "converter": item.converter,
    "data": item.data,
    "processingPath": item.processingPath or ""
})
```

Die äussere Archiv-id ist der SQLite-Schlüssel. payload.wamas_message_id ist die ursprüngliche WAMAS-id. Beispiel: Archiv-id 4 gehört zu WAMAS-id 505. Das gesamte Archivobjekt darf deshalb nicht unverändert an BC gesendet werden. [Q1, Q2]

Antwort des Referenzreceivers

```
errors = []
# Pro Item speichern; bei Ausnahme errors.append({"id": ..., "msg": ...})
return {"items": errors}
```

Bei vollständiger Speicherung liefert der Code {"items":[]} mit dem Standardstatus 200. Bei einzelnen Speicherfehlern enthält items die betroffenen IDs und msg; der HTTP-Status bleibt in diesem Code ebenfalls 200. Die Speicherung erfolgt pro Item in einer eigenen Datenbanktransaktion. Das ist eine Referenzimplementierung und keine verbindliche WAMAS-Protokollspezifikation. [Q1]

Der NAS-Eingang verwendet HTTP Basic; BC wurde mit OAuth2 Bearer getestet. Eine Bestätigung der Archivierung bedeutet im NAS noch keine fachliche ERP-Buchung. Der Archivstatus PROCESSED wird über eine separate Route gesetzt. [Q1]

5 Echte WAMAS-Beispiele

Die folgenden fachlichen Ausschnitte stammen aus dem NAS-Export mit 35 Datensätzen. Alle drei gewählten Nachrichten verwenden JsonConverter, einen leeren processingPath und die Richtung WAMAS → HOST. Die Ausschnitte lassen Leerfelder und weitere Strukturen weg und sind keine vollständigen Sendebodies. Vollständige rekonstruierte Requests liegen im ergänzenden Testpaket. [Q2]

Telegramm	WAMAS / Archiv	Transaktion und Archivzeit UTC
WLSSTOCKADJ001	505 / 4	505-1 · 21.08.2026 13:38:54
WLSPDACK002	516 / 15	516-1 · 22.08.2026 19:41:26
WLSSDACK002	517 / 16	517-1 · 22.08.2026 19:41:32

WLSSTOCKADJ001 mit einer Bestandsänderung

```
"WLSSTOCKADJ001": [{
  "clientId": "DEFAULT", "itemNo": "20082026",
  "variant": "00000", "baseQty": -9,
  "luId": "18888101001-021-001",
  "stoLoc": "A1-OP1-1", "postingKey": "502"
}]
```

Der Originalheader enthält HEADER_TRANSACTIONID="505-1" und HEADER_CREATIONTIME="20260821153634". Dieser WAMAS-Zeitstempel enthält keine explizite Zeitzone; er wird unverändert wiedergegeben. Genau diese Nachricht wurde für den erneuten BC-POST herangezogen. [Q2, B4]

WLSPDACK002 als Auftragsrückmeldung

```
"demandNo": "WLSPD20082027", "typeId": "SHPG",
"state": "FINISHED",
"WLSPDLACK001": [{
  "itemNo": "20082026", "diffReason": "NONE",
  "orderedAmount": 1, "releasedAmount": 1, "deliveredAmount": 1
}]
```

Im vollständigen Original ist zusätzlich WLSPDLACTIVITYACK001 enthalten, unter anderem mit pickedAmount=1, reservedAmount=1 und sourceLocation="A1-OP1-1". Ein erfolgreicher BC-POST dieses vollständigen Typs ist noch nicht belegt. [Q2]

WLSSDACK002 als Auftragsrückmeldung

```
"demandNo": "20082026", "typeId": "RECV",
"state": "FINISHED",
"WLSSDLACK001": [{
  "itemNo": "20082026", "diffReason": "NONE",
  "orderedAmount": 4, "releasedAmount": 4, "deliveredAmount": 4
}]
```

Das vollständige Original enthält WLSSDLACTIVITYACK001 mit storedAmount=4 und reservedAmount=4. Diese Archivnachricht mit ID 517 ist vom gekürzten Test zu IR000007 und Transaction 1003-1 zu unterscheiden. Das erste Deep-Insert-Screenshot zeigt den gekürzten Test, nicht die vollständige Archivnachricht 517. [Q2, B3]

6 Sollvertrag und Zielarchitektur

SOLL: WAMAS sendet einen einzigen nativen Batch-POST. Der Empfänger übernimmt die technische BC-intern benötigte Datensatzbildung. WAMAS muss dafür weder einen Parent vorab anlegen noch eine BC Entry No., Parent-GUID oder source mitsenden.

WAMAS Lift & Store

- | 1 Token holen: Client Credentials → Microsoft Entra ID
- | 2 HTTPS POST + Bearer + natives items-Array

v

BC-seitiger WAMAS-Empfänger

- | Token / Company / Request prüfen
- | Parent intern anlegen und gültigen Schlüssel übernehmen
- | Jedes Item dem erzeugten Parent zuordnen
- | Nutzdaten dauerhaft speichern / Transaktion abschliessen

v

HTTP 2xx + vereinbarte WAMAS-Antwort

|

v

WAMAS bestätigt den Versand

BC-Verarbeitung → fachliche Buchung / Fehlerstatus / Monitoring

Abbildung 1 – Empfohlene Zielarchitektur. Dauerhafte Annahme und spätere fachliche Verarbeitung sind getrennte Zustände. Ein NAS ist im Zielpfad nicht erforderlich; es dient als Referenz für Format und historische Nutzdaten.

Request in Postman

POST {{bcBase}}/api/acumax/wamas/v2.0/wamasInboundMessages

Params: company = AcuMax Med AG

Authorization: Bearer <aktueller Access Token>

Accept: application/json

Content-Type: application/json

```
{
  "items": [{
    "id": 505,
    "converter": "JsonConverter",
    "data": "<vollständiges DI_TELEGRAMS-JSON als String>",
    "processingPath": ""
  }]
}
```

Der Platzhalter oben erläutert die Struktur und ist nicht sendefähig. Für die Wiederholung ist request_505_WLSSTOCKADJ001.json aus dem Testpaket zu verwenden. data bleibt ein String: Anführungszeichen des inneren JSON sind im äusseren JSON escaped. Keine zusätzliche Kodierung über diesen JSON-String hinaus vornehmen. In Postman unter Params den Namen ohne %20 eingeben; Postman übernimmt das URL-Encoding.

Erfolgsantwort und ihre Bedeutung

HTTP/1.1 200 OK

Content-Type: application/json

```
{"items":[]}
```

SOLL, an Q1 orientiert: 200 mit leerer Fehlerliste nach dauerhaft erfolgreicher Annahme. Andere 2xx-Codes, insbesondere 201 oder 202, sind erst nach Prüfung des konkreten WAMAS-Senders freizugeben. Eine generische OData-Entität in einer 201-Antwort ist nicht automatisch WAMAS-kompatibel. Der reale Sender muss Status und Body als Erfolg auswerten. Die Antwort darf nicht schon Erfolg melden, solange Nutzdaten nur flüchtig im Speicher liegen.

7 Umsetzung auf Business Central Seite

Vorhandenes Datenmodell und Prüfpunkte

GESPRÄCH: Die Metadaten beschreiben `wamasInboundMessages` mit `id`, `entryNo`, `status`, `source` und der `Navigation items`. `source` erlaubt REST oder File. Die Items enthalten `systemId`, `messageId`, `id`, `converter`, `data` und `processingPath`; `data` ist `Edm.String`. Der `Parent-EntitySet` wurde mit `Insertable=true`, die separaten Items mit `Insertable=false` beschrieben. [G]

Das Vorhandensein der `Navigation` und ein erfolgreicher `$expand-GET` beweisen keine funktionierende verschachtelte Schreibverarbeitung. Auch `Insertable=false` am separaten `Child-EntitySet` entscheidet allein nicht, ob ein `Deep Insert` über den `Parent` unterstützt wird. Die tatsächliche Implementierung und ihre Metadaten müssen zusammen geprüft werden.

Verbindlich herzustellendes Verhalten

SOLL: Der Empfänger akzeptiert `items[]` direkt. Er erzeugt einen `Parent` innerhalb des `Requests`, übernimmt dessen tatsächlich gespeicherten Schlüssel und verknüpft alle angenommenen Items damit. WAMAS-id und BC-interne Schlüssel bleiben getrennt. `source=REST` wird bei Bedarf intern gesetzt. Der Fehlerpfad darf keinen Lookup auf einen noch nicht erzeugten `Parent` mit `Entry No.=0` ausführen.

SOLL: Der ERP-Partner prüft `API-Page`, `Page-Part`, `SourceTable` beziehungsweise temporären Buffer, `Insert-/Validate-Trigger`, `Nummernvergabe` und `Schlüsselerückgabe`. Für den konkreten Fehler sind die `CorrelationIds` aus B3 und B4 mit BC-Telemetrie oder Debugger zu verfolgen. Die Korrektur ist durch ein anschliessendes Lesen der persistierten `Parent-/Child-Beziehung` nachzuweisen.

Zwei mögliche Implementierungswege

Variante A – Angepasste Custom API mit `Deep Insert`. Microsoft dokumentiert die gemeinsame Anlage von `Parent` und `Children` in einem `POST`. Für die Modellierung empfiehlt Microsoft `SystemId` als `ODataKeyFields` sowie die konsistente Abbildung der Beziehungsfelder in `API-Pages` und `SubPageLink`. Der ERP-Partner muss zusätzlich die `Trigger-` und `Antwortlogik` des konkreten WAMAS-Endpunkts prüfen. Diese Eigenschaften allein garantieren keine Fehlerbehebung. [M2]

Konzept, kein direkt übernehmbarer AL-Code:

```
Parent API Page: PageType = API
                  ODataKeyFields = SystemId
                  DelayedInsert = true
Part items:      SubPageLink = Parent SystemId → Parent.SystemId
Child:           gültige Parent-Referenz vor Speicherung
```

Variante B – Spezifische Empfangsschicht. Der ERP-Partner stellt einen Empfänger bereit, der den WAMAS-Vertrag annimmt und intern die BC-Datensätze erzeugt. Falls die BC-OData-Oberfläche das erforderliche `Request-/Response-Format` nicht direkt bereitstellen kann, muss der Partner einen technisch unterstützten `Service-/Adapterweg` vorlegen. Ein frei gestaltbarer `REST-Body` und eine beliebige `JSON-Antwort` dürfen für eine normale `AL-API-Page` nicht vorausgesetzt werden. Eine zusätzliche Komponente wäre eine ausdrücklich zu vereinbarende Architekturänderung.

Transaktionen und Fehlerantworten

SOLL: Vor Umsetzung festlegen, ob ein `Batch` vollständig zurückgerollt wird oder einzelne Items angenommen werden. Der `NAS-Receiver` arbeitet pro Item; sein Fehlerbeispiel lautet `{"items":[{"id":505,"msg":"Message could not be stored: ..."}]}`. BC darf dieses Format erst nach Abgleich mit der verbindlichen WAMAS-DI-API-Version übernehmen. Bei Teilfehlern müssen `Sender` und `Empfänger` wissen, welche IDs erneut zu senden sind. Technische Fehler dürfen nicht als leere Erfolgsliste ausgegeben werden.

SOLL: Wiederholungen müssen dieselbe fachliche Wirkung höchstens einmal auslösen. Dazu einen stabilen Schlüssel aus `Senderkontext` und WAMAS-id beziehungsweise `Telegramm-Transaktion` vereinbaren; `Company`, `Neustarts` und `ID-Wiederverwendung` berücksichtigen. Persistierte Annahme, fachliche Verarbeitung und fachlicher Fehler müssen getrennt nachvollziehbar sein.

8 Übergabe und Abnahmekriterien

Verantwortung	Konkrete Lieferung
ERP-/BC-Partner	Korrigierte Extension bzw. Empfänger; genaue URL; Request-/Response-Vertrag; BC-App-Berechtigungen; Parent-/Child-Nachweis; Fehler- und Dublettenlogik; Telemetrieauswertung.
SSI/WAMAS	Version und DI-API-Spezifikation; OAuth-/OIDC-Konfiguration einschliesslich JWK; Senderregeln für Statuscode und Body; Retry, Timeout, Batchgrösse; Versandnachweise.
Gemeinsam	Testdaten und erwartete fachliche Ergebnisse festlegen; Testfälle unten durchführen; Rohresponses ohne Tokens sichern; Freigabe für Produktivbetrieb dokumentieren.

BC-Zugang: Die Anwendung ist für die Ziel-Company zu aktivieren und mit den notwendigen API-/Tabellen-/Codeunit-Rechten auszustatten. Microsoft beschreibt dafür Entra-Anwendungsberechtigungen und die zusätzliche Freigabe der App in BC; die erforderlichen Objektberechtigungen sind vom Partner zu bestätigen. Der erfolgreiche Testzugriff ersetzt keine dokumentierte Betriebskonfiguration. [M1]

Nr	Abnahmetest	Bestanden wenn
A01	OAuth2 direkt aus WAMAS	Tokenabruf und Erneuerung funktionieren; korrekter Tenant/Scope; JWK-Frage geklärt.
A02	Native Requests 505, 516 und 517	Alle drei vollständigen Telegrammtypen werden ohne vorgeschalteten Parent-POST angenommen; vereinbarter 2xx und gültiger Response-Body.
A03	Parent und Items kontrollieren	Gültiger Parent gespeichert; alle Items korrekt zugeordnet; data unverändert; keine Entry-No.-0-Abhängigkeit.
A04	Echter WAMAS-Sender	WAMAS bewertet dieselbe Antwort als Versand erfolgreich; kein unbegründeter Retry; BC-Nachricht auffindbar.
A05	Fachliche Verarbeitung	Für alle drei Typen stimmen Auftrag/Artikel, Mengen, Richtung und Status mit vorher vereinbarten Sollwerten überein.
A06	Mehrere Items und Teilfehler	Vereinbarte Atomarität eingehalten; Fehler eindeutig einer ID zugeordnet; Wiederholung erzeugt keine Doppelbuchung.
A07	Timeout nach Speicherung und Dublette	Wiederholung wird sicher erkannt; gleiche Nachricht hat keine zweite fachliche Wirkung.
A08	Fehlende Company und ungültiger Token	Definierte Fehlerantwort; keine Speicherung in falscher Company; keine positive Leerantwort bei Fehler.
A09	Ungültige Daten und unbekannter Converter	Kontrollierter Fehler statt Datenverlust; WAMAS-Retry/Fehlerzustand entspricht dem vereinbarten Vertrag.
A10	Betrieb und Lastgrenzen	Maximale Payload-/Batchgrösse, Timeout, 429/5xx-Verhalten, Monitoring und Wiederanlauf dokumentiert und geprüft.

Freigabenachweis je Test: Umgebung und Extension-Version, WAMAS-Version, Datum/Uhrzeit mit Zeitzone, Request-Datei und Hash, WAMAS-id/Transaktions-ID, HTTP-Code, bereinigter Response, BC-Parent-/Child-Identität und Ergebnis im WAMAS-Sender. Abnahme erst nach gemeinsamer Bestätigung; diese Dokumentation ist keine Produktivfreigabe.

9 Quellen und Belegregister

Die Belege sind durch ihre Kennungen in Text und Testmatrix referenziert. Screenshots werden unverändert eingebettet. Die ausgewählten Bilder enthalten keine sichtbaren Client Secrets, Passwörter oder Bearer Tokens. Zugangsdatenbilder sind nicht enthalten.

ID	Quelle und Beweisumfang
G	Gespräch „REWST API Verbindung testen“, 23.09.2026. ID 6ab38ace-2fd4-83eb-98e0-0781aafbfbf7. Chronologie, Token-/Minimal-POST-Ergebnisse und Metadatenbeschreibung. Company-Liste als wörtlicher Nutzertext.
Q1	main.py, FastAPI-Receiver Version 3.3.0. Modelle, DI-POST-Route, Archiv-Mapping, Basic Auth, Speicher- und Antwortlogik direkt geprüft.
Q2	Pasted markdown.md, NAS-Export mit 35 Datensätzen. Ausgewählt: Archiv-id 4/15/16 mit WAMAS-id 505/516/517. Vollständige data-Strings direkt geprüft.
B1	Original-Screenshot: GET ohne Company, 404 Internal_CompanyNotFound. Anhang A.
B2	Original-Screenshot: Kontroll-GET mit Company und expand, 200 OK, entryNo 2 und 1 mit leeren Items. Anhang B.
B3	Original-Screenshot: source plus items, gekürztes WLSSDACK002, 404 Internal_RecordNotFound. Anhang C.
B4	Original-Screenshot: natives items-Format mit WAMAS-id 505, 404 Internal_RecordNotFound. Anhang D.

Einordnung der Nachweise

Token 200, anfängliche 401, Minimal-POST 201 und die Detailangaben zu \$metadata sind hier als Gesprächsnachweise eingeordnet. Die frühen Originalbilder und das ursprüngliche Metadaten-XML sind nicht Teil des eingebetteten Belegsatzes. Für eine revisionsfähige Abnahme sind bereinigte Originalresponses beziehungsweise das aktuelle Metadaten-XML zu sichern. Die Postman-Screenshots zeigen Tests gegen BC; sie ersetzen keinen End-to-End-Nachweis aus WAMAS.

Herstellerreferenzen zur Implementierung

M1 – Microsoft Learn, Using service-to-service (S2S) authentication. Grundlage für Client Credentials, Scope, Token-URL und App-Freigabe in BC. Abruf 23.09.2026.

<https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/administration/automation-apis-using-s2s-authentication>

M2 – Microsoft Learn, Developing a custom API. Grundlage für Deep Insert und die Modellierung über SystemId und SubPageLink. Abruf 23.09.2026.

<https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-develop-custom-api>

Die Herstellerreferenzen erklären technische Möglichkeiten; sie beweisen nicht die Implementierung der kundenspezifischen ACU-Extension. Die verbindliche WAMAS-DI-API-Spezifikation ist durch SSI für die eingesetzte Version zu ergänzen.

Reproduzierbare Nutzdaten im Testpaket

Die drei request_*.json-Dateien rekonstruieren den äusseren Batch aus dem Archiv. Dabei wird nur payload.wamas_message_id zurück auf items[].id abgebildet. converter, data und processingPath bleiben inhaltlich unverändert. Es handelt sich nicht um gespeicherte HTTP-Mitschnitte. Die Dateien enthalten keine Authentifizierungsdaten. Ein Manifest enthält die SHA-256-Prüfsummen.

Anhang A Company muss angegeben werden

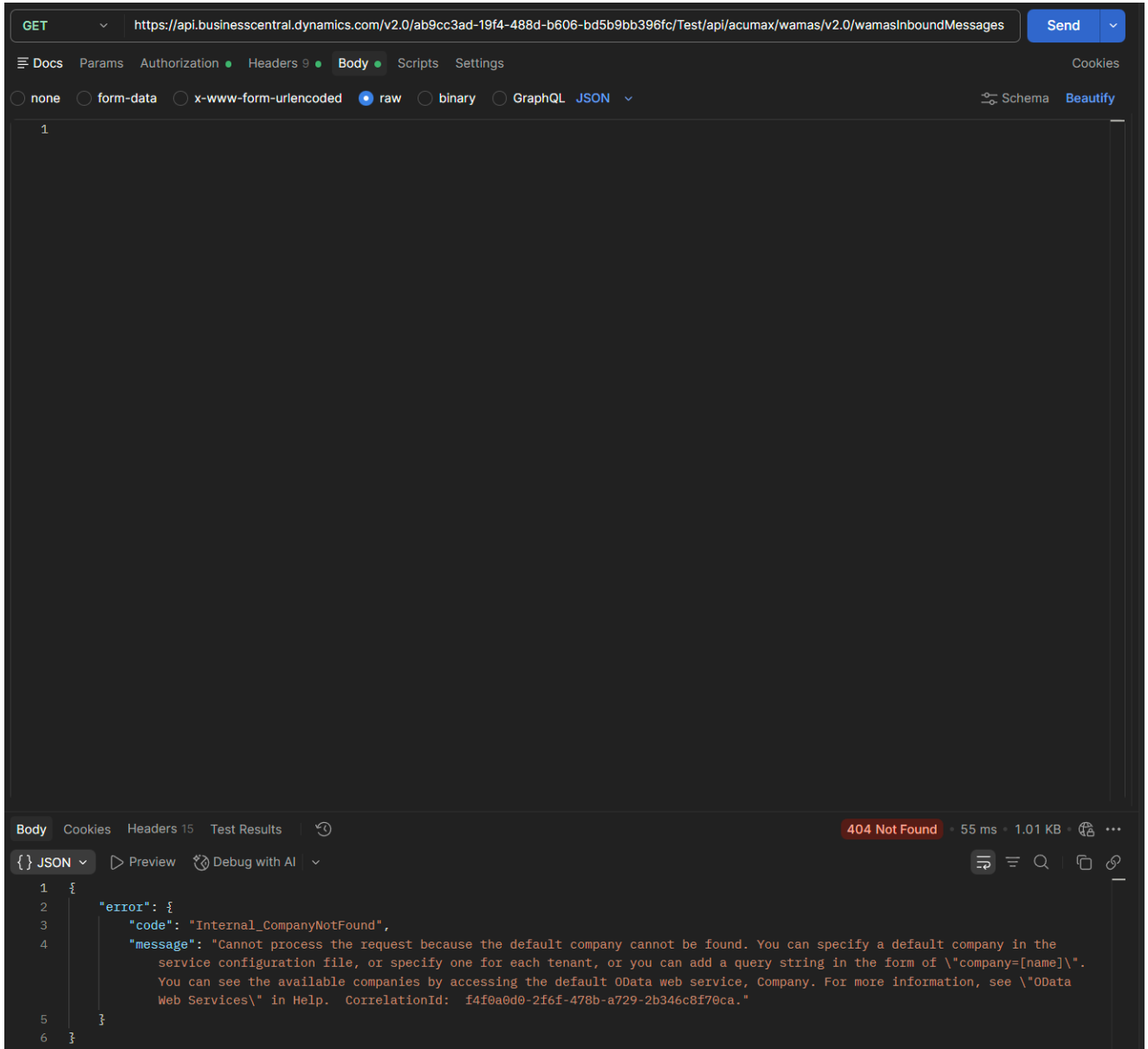


Abbildung 2 – B1. GET ohne Company mit 404 Internal_CompanyNotFound. Die Antwort fordert eine Company-Auswahl. Dieser Screenshot belegt den später wiederholten Company-Fehler, nicht den anfänglichen 401.

Anhang B Lesen mit Company funktioniert

The screenshot shows a REST client interface with the following details:

- URL:** `https://api.businesscentral.dynamics.com/v2.0/ab9cc3ad-19f4-488d-b606-bd5b9bb396fc/Test/api/acumax/wamas/v2.0/wamasInbou`
- Method:** GET
- Query Params:**

Key	Value	Description	Bulk Edit	...
company	AcuMax Med AG			
\$orderby	entryNo desc			
\$top	5			
\$expand	Items			
- Response:** 200 OK, 484 ms, 1.08 KB
- Body (JSON):**

```

1 {
2   "@odata.context": "https://api.businesscentral.dynamics.com/v2.0/ab9cc3ad-19f4-488d-b606-bd5b9bb396fc/Test/api/acumax/wamas/v2.0/$metadata#wamasInboundMessages",
3   "value": [
4     {
5       "@odata.etag": "W/\"JzIwOzEyNDYzOTYzODQwMzM0ODAyMzAwMTswMDsn\"",
6       "id": "cdb0532a-cdb5-f111-aaa8-6045bd2c3231",
7       "entryNo": 2,
8       "status": "Message received",
9       "source": "REST",
10      "items": []
11    },
12    {
13      "@odata.etag": "W/\"JzE5Ozg2Njk5MTYwOTYzNDQzNzYxNDcxOzAwOyc=\"",
14      "id": "6b7dd6d8-ccb5-f111-aaa8-6045bd2c3231",
15      "entryNo": 1,
16      "status": "Message received",
17      "source": "REST",
18      "items": []
19    }
20  ]
21 }

```

Abbildung 3 – B2. Kontroll-GET mit company=AcuMax Med AG, Sortierung, top und expand. HTTP 200; sichtbar sind entryNo 2 und 1 mit jeweils leerem items-Array.

Anhang C Erster Deep Insert schlägt fehl

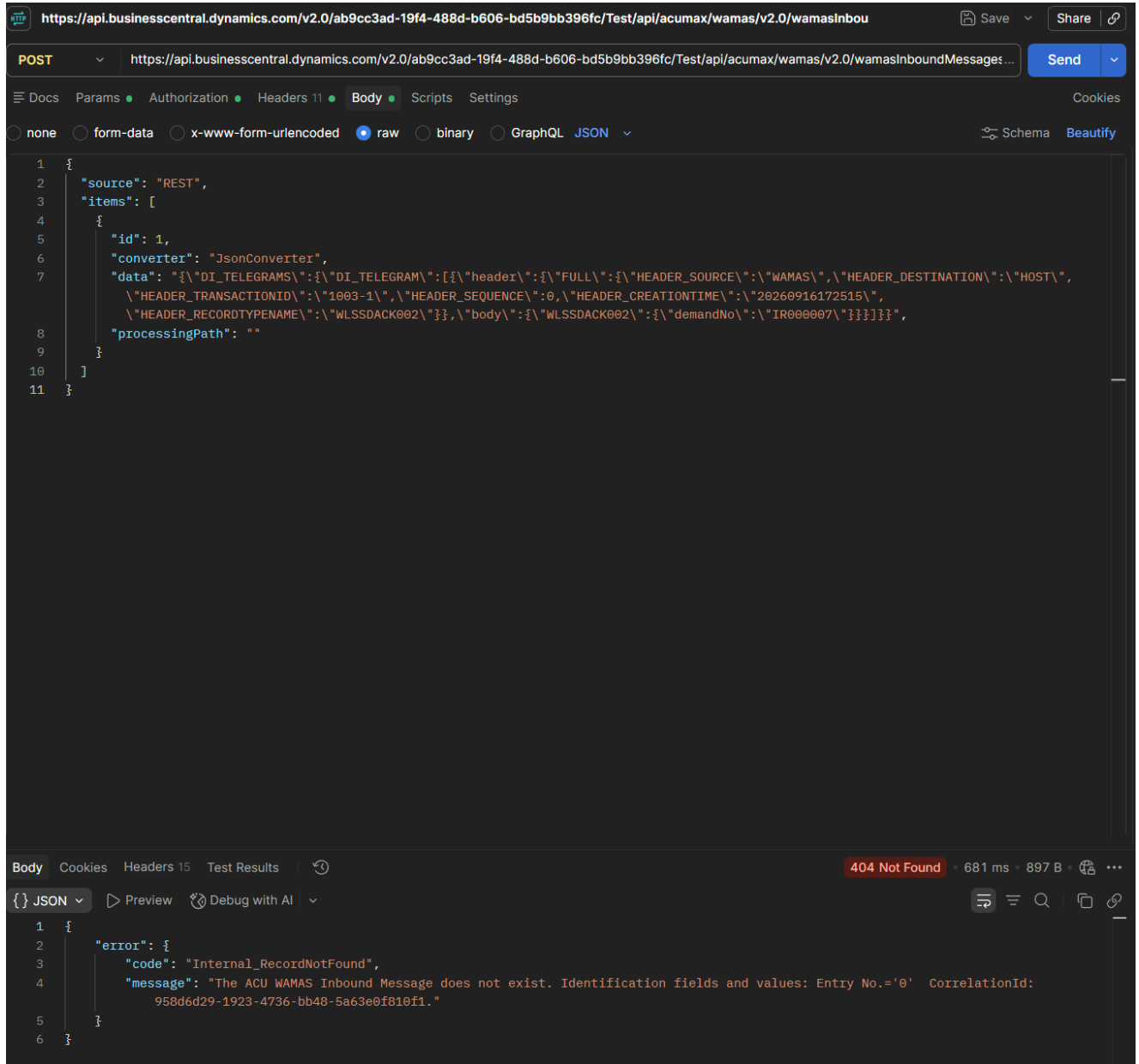


Abbildung 4 – B3. POST mit source=REST und einem gekürzten WLSACK002. HTTP 404 Internal_RecordNotFound, Entry No.=0. CorrelationId: 958d6d29-1923-4736-bb48-5a63e0f810f1.

Anhang D Nativer Payload bestätigt den Fehler

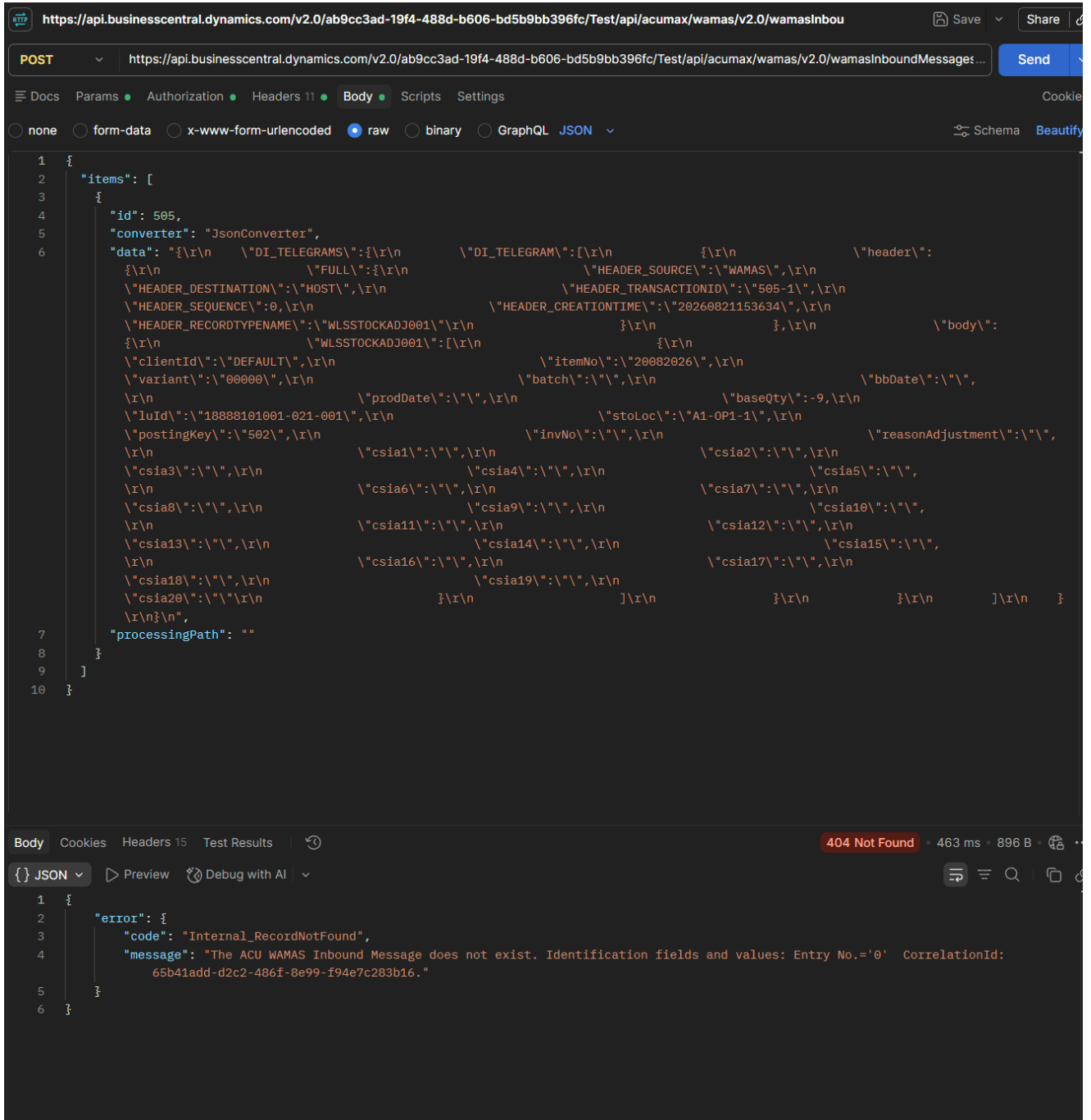


Abbildung 5 – B4. POST ohne source mit WAMAS-id 505 und WLSSTOCKADJ001. Erneut HTTP 404 Internal_RecordNotFound, Entry No.=0. CorrelationId: 65b41add-d2c2-486f-8e99-f94e7c283b16.