

TECHNICAL DOCUMENTATION

WAMAS Lift and Store to Business Central

REST and OAuth2
Test evidence and receiver requirements

AcuMax Med AG | Test environment | Version 1.0 EN | 23 September 2026

Findings and required action

Access to the Business Central custom API was established in Postman. The conversation records a successful OAuth2 token request, a successful GET and a minimal POST returning HTTP 201. However, the decisive POST with a WAMAS payload reconstructed from the NAS archive was rejected with HTTP 404 and Internal_RecordNotFound. The error identifies the missing ACU WAMAS Inbound Message record with Entry No.=0. [G, B4]

Direct receipt of complete WAMAS messages in Business Central has therefore not yet passed acceptance testing. The ERP partner must implement or correct acceptance of the native WAMAS format, internal parent creation, item assignment and the response to the WAMAS sender. A preliminary parent POST by WAMAS is not part of the evidenced reference contract. [Q1, Q2]

Purpose and audience

This handover is intended for the ERP/Business Central partner and SSI/WAMAS technicians. It describes the test sequence, classifies the evidence and defines the target interface contract and verifiable acceptance criteria. Production approval requires a successful test with the actual WAMAS sender.

Reading guide

Section	Contents
1 to 3	Connection details, chronological test sequence and test matrix
4 to 5	Receiver code and real WAMAS examples
6 to 8	Target architecture, BC implementation and acceptance
9 and appendices	Source register and four screenshot exhibits

Evidence levels: DIRECT = screenshot, file or verbatim user-provided output inspected. CONVERSATION = result recorded in the earlier conversation. ANALYSIS = technical inference. TARGET = implementation recommendation or outstanding acceptance item. These labels distinguish observed results from confirmed root causes.

1 Connection details and OAuth2

Parameter	Value used in the 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 · Group wamas · Version v2.0
EntitySet	wamasInboundMessages
Scope	https://api.businesscentral.dynamics.com/.default
Grant Type	client_credentials
Client secret and access token	Excluded from this document; configure using protected storage.

The company ID comes from the companies response pasted by the user. The successful custom API request uses the company name as a query parameter. A call to this custom API using the company GUID is not evidenced in the test sequence. [G, B2]

Endpoints

The request examples below use {{bcBase}} for readability:

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

The lines above form a single URL without a line break. 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
```

Requesting a token 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 = <securely stored secret value>
scope        = https://api.businesscentral.dynamics.com/.default
```

CONVERSATION: The token test returned 200 OK, token_type=Bearer and expires_in=3599. This confirms token issuance at that time, but does not by itself prove BC object permissions or payload processing. Use the token lifetime returned in each actual response. [G, M1]

OPEN: The supplied JWK URL points to tenant ce896e4e-701d-4691-8bbb-e7059c7dc33d, which differs from the tenant in the token and BC URLs. It was not used in the manual Postman test. SSI and the ERP partner must confirm the appropriate discovery/JWK configuration for the deployed WAMAS version before commissioning. [G]

2 Chronological test sequence

The Postman test sequence took place on 23 September 2026. The times below identify the corresponding conversation entries in Europe/Zurich (UTC+2), not independently measured server timestamps. The NAS messages had already been archived in August and September 2026. [G, Q2]

Authentication and company selection

10:16 to 10:24 - The connection details supplied by the ERP partner were configured in Postman. The separate client credentials POST to Microsoft Entra returned 200 OK with a bearer token. [G; CONVERSATION]

Next - The initial BC GETs returned 401 Unauthorized with "The credentials provided are incorrect". The exact cause is not conclusively established. A missing or untransferred bearer token was suspected for the new companies request; no actual permissions change by the ERP partner is documented. A missing Accept header alone is not an evidenced cause. [G; CONVERSATION/ANALYSIS]

10:32 to 10:41 - After the authentication step, 404 Internal_CompanyNotFound appeared. The companies query identified AcuMax Med AG and its company ID. The subsequent GET on wamasInboundMessages succeeded with company=AcuMax Med AG. The company error later recurred on a GET without a company and is directly evidenced by a screenshot. [G, B1, B2]

Data model inspection and write tests

10:44 to 10:48 - The custom API metadata was examined, followed by a query of the existing parents using \$expand=items. The navigation property was readable, but the item collections were empty. [G; CONVERSATION]

10:54 - The minimal POST with {"source":"REST"} returned 201 Created. However, the response contained a zero GUID, entryNo=0 and an empty status. This response alone therefore confirms neither a processed WAMAS message nor correct return of the persisted record key. [G; CONVERSATION]

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

11:02 - A deep insert with source=REST and items[0] failed with 404 Internal_RecordNotFound, Entry No.=0. data contained an abbreviated WLSSDACK002 for IR000007; this was a structural test, not a complete WAMAS business processing test. [B3; DIRECT]

11:07 - A verification GET with company, \$orderby=entryNo desc, \$top=5 and \$expand=items returned 200 OK. It showed entryNo 2 and 1, status "Message received", source "REST" and items=[] for both. No additional visible parent from the failed deep insert was found. This alone does not prove a complete transaction rollback. [B2; DIRECT/ANALYSIS]

Comparison with real WAMAS data

11:16 to 11:23 - The NAS archive and then main.py clarified the outer request structure. The interim hypothesis that WAMAS sends a single item without an items array was corrected by inspection of the receiver code. [G, Q1, Q2]

11:29 - The repeated POST without source, using WAMAS ID 505 and the archived WLSSTOCKADJ001 as the data string, again returned 404 Internal_RecordNotFound, Entry No.=0. The native structure reconstructed from real data therefore reproduces the same error. This was still not a direct WAMAS-to-BC execution. [B4; DIRECT]

3 Test matrix and error assessment

Test	Request and check	HTTP	Result and evidence
T01	Token POST with client credentials	200	Token issued; Bearer, 3599 s. CONVERSATION [G].
T02	Initial GETs to BC or companies	401	Access rejected. Exact cause open; missing token transfer is a hypothesis. [G]
T03	GET without an effective company selection	404	Internal_CompanyNotFound. Company missing or not resolved. DIRECT [B1].
T04	GET standard API companies	Success ¹	Company name and GUID evidenced in original text. DIRECT [G].
T05	GET custom API with company	200	Parents readable. Initial success recorded in conversation; verification GET directly evidenced. [G, B2]
T06	GET custom API \$metadata	XML ¹	Parent and child fields documented in the conversation. [G]
T07	GET with \$expand=items	200	Two parents; both item collections empty. DIRECT [B2].
T08	POST {"source":"REST"}	201	Minimal POST accepted; response values zero GUID/0/empty. CONVERSATION [G].
T09	POST source and items with abbreviated WLSSDACK002	404	Internal_RecordNotFound; parent Entry No.=0. DIRECT [B3].
T10	Verification GET after T09	200	Still entryNo 2 and 1, with empty items. DIRECT [B2].
T11	Inspect NAS code and archive	—	Schema and original payloads evidenced; no new HTTP test. [Q1, Q2]
T12	POST native items format with original data from ID 505	404	Same parent error even without source. DIRECT [B4].

¹ Successful data/metadata retrieval is documented; a separately verifiable status code is not available for these two steps. No 200 code is added merely because it would be expected.

Distinguishing three error categories

401 concerns access to the service. Internal_CompanyNotFound concerns company selection.

Internal_RecordNotFound naming the ACU table concerns record processing inside the custom API. Here, a 404 does not automatically mean that the URL or server does not exist.

ANALYSIS: Both failed POSTs direct the investigation towards the parent record lifecycle and child assignment. The exact AL statement, trigger sequence and any use of temporary records are not established without BC code and telemetry. Likewise, the early parent error does not prove the business validity of every telegram field.

4 Reference receiver and native schema

DIRECT: main.py defines the WAMAS entry point as POST /di-api/v3/HOST/messages. The separate /api/v3/messages route has a different POST contract: it expects type and payload; a GET on that route returns the archive. [Q1]

Models defining the outer request

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

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

The code evidences the accepted batch schema; the archive evidences the stored real WAMAS content. Together they allow reconstruction of the native request structure. They are not an unmodified HTTP capture including headers and the original byte sequence. In particular, the Python model accepts any data value; in the selected real messages, data is a JSON string. [Q1, Q2]

Field	Evidenced meaning
items	Outer list containing one or more WamasItem objects.
id	WAMAS message ID; Python accepts int or str. Not a BC Entry No.
converter	JsonConverter in the archive. A BC converter must support this value.
data	Complete DI_TELEGRAMS JSON as a string in the archive examples.
processingPath	Empty in the archive. Receiver normalises None or empty values to "".

Archiving changes the field name

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

The outer archive id is the SQLite key. payload.wamas_message_id is the original WAMAS id. For example, archive id 4 corresponds to WAMAS id 505. The entire archive object must therefore not be sent to BC unchanged. [Q1, Q2]

Reference receiver response

```
errors = []
# Store each item; on exception errors.append({"id": ..., "msg": ...})
return {"items": errors}
```

When all items are stored, the code returns {"items":[]} with the default status 200. For individual storage failures, items contains the affected IDs and msg; this code still returns HTTP 200. Each item is stored in its own database transaction. This is a reference implementation, not a binding WAMAS protocol specification. [Q1]

The NAS entry point uses HTTP Basic; BC was tested with OAuth2 Bearer. Acknowledging archive storage does not imply a business posting in the ERP. The archive status PROCESSED is set through a separate route. [Q1]

5 Real WAMAS examples

The following business data excerpts come from the NAS export containing 35 records. All three selected messages use JsonConverter, an empty processingPath and the direction WAMAS → HOST. The excerpts omit empty fields and additional structures; they are not complete request bodies. Complete reconstructed requests are provided in the supplementary test package. [Q2]

Telegram	WAMAS / Archive	Transaction and archive time in UTC
WLSSTOCKADJ001	505 / 4	505-1 · 21 Aug 2026 13:38:54
WLSPDACK002	516 / 15	516-1 · 22 Aug 2026 19:41:26
WLSSDACK002	517 / 16	517-1 · 22 Aug 2026 19:41:32

WLSSTOCKADJ001 with a stock adjustment

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

The original header contains HEADER_TRANSACTIONID="505-1" and HEADER_CREATIONTIME="20260821153634". This WAMAS timestamp has no explicit time zone and is reproduced unchanged. This is the message used for the repeated BC POST. [Q2, B4]

WLSPDACK002 as an order acknowledgement

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

The complete original also contains WLSPDLACTIVITYACK001, including pickedAmount=1, reservedAmount=1 and sourceLocation="A1-OP1-1". A successful BC POST of this complete telegram type has not yet been evidenced. [Q2]

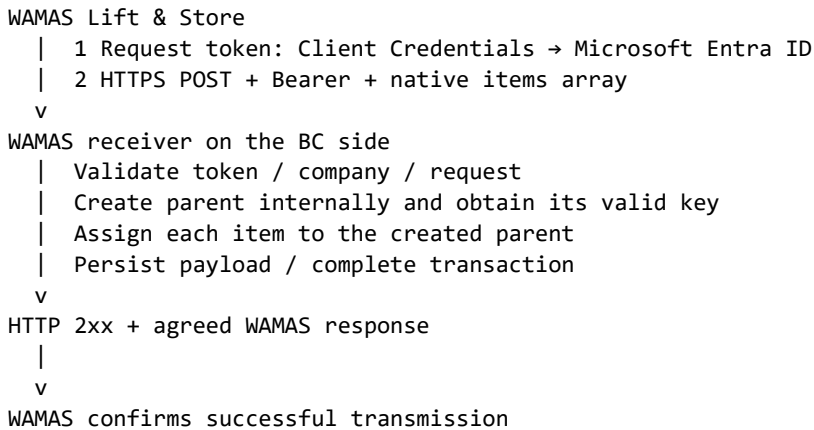
WLSSDACK002 as an order acknowledgement

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

The complete original contains WLSSDLACTIVITYACK001 with storedAmount=4 and reservedAmount=4. This archived message with ID 517 must be distinguished from the abbreviated test for IR000007 and transaction 1003-1. The first deep insert screenshot shows the abbreviated test, not the complete archived message 517. [Q2, B3]

6 Target contract and architecture

TARGET: WAMAS sends one native batch POST. The receiver creates the records required internally by BC. WAMAS must not have to create a parent beforehand or supply a BC Entry No., parent GUID or source.



BC processing → business posting / error status / monitoring

Figure 1 - Recommended target architecture. Durable acceptance and subsequent business processing are separate states. A NAS is not required in the target path; it serves as a reference for the format and historical payloads.

Request in Postman

```

POST {{bcBase}}/api/acumax/wamas/v2.0/wamasInboundMessages
Params: company = AcuMax Med AG
Authorization: Bearer <current access token>
Accept: application/json
Content-Type: application/json
  
```

```

{
  "items": [{
    "id": 505,
    "converter": "JsonConverter",
    "data": "<complete DI_TELEGRAMS JSON as a string>",
    "processingPath": ""
  }]
}
  
```

The placeholder above illustrates the structure and must not be sent as a real payload. For the repeat test, use request_505_WLSSTOCKADJ001.json from the test package. data remains a string: quotation marks in the inner JSON are escaped in the outer JSON. Do not apply additional encoding beyond this JSON string. In Postman Params, enter the company name without %20; Postman handles URL encoding.

Success response and its meaning

```

HTTP/1.1 200 OK
Content-Type: application/json
  
```

```

{"items":[]}
  
```

TARGET, based on Q1: 200 with an empty error list after successful durable acceptance. Other 2xx codes, especially 201 or 202, require validation with the actual WAMAS sender before approval. A generic OData entity in a 201 response is not automatically WAMAS-compatible. The actual sender must interpret both the status and body as success. The response must not indicate success while payloads exist only in volatile memory.

7 Implementation on the Business Central side

Existing data model and checks

CONVERSATION: The metadata describes `wamasInboundMessages` with `id`, `entryNo`, `status`, `source` and the items navigation property. `source` allows REST or File. Items contain `systemId`, `messageId`, `id`, `converter`, `data` and `processingPath`; `data` is `Edm.String`. The parent entity set was described as `Insertable=true` and the separate items as `Insertable=false`. [G]

The existence of the navigation property and a successful \$expand GET do not prove that nested writes work. `Insertable=false` on the separate child entity set alone does not determine whether a deep insert through the parent is supported. The implementation and its metadata must be examined together.

Required target behaviour

TARGET: The receiver accepts `items[]` directly. It creates a parent within the request, obtains its actual persisted key and links all accepted items to it. The WAMAS id and BC internal keys remain separate. `source=REST` is set internally if required. The error path must not look up a parent with `Entry No.=0` before that parent has been created.

TARGET: The ERP partner checks the API page, page part, `SourceTable` or temporary buffer, insert/validate triggers, number assignment and key return. For the specific error, trace the `CorrelationIds` from B3 and B4 in BC telemetry or the debugger. Prove the correction by subsequently reading the persisted parent/child relationship.

Two possible implementation approaches

Option A - Adapted custom API with deep insert. Microsoft documents creating a parent and its children in one POST. Microsoft recommends `SystemId` as `ODataKeyFields` and consistent exposure of relationship fields in API pages and `SubPageLink`. The ERP partner must also inspect the trigger and response logic of the specific WAMAS endpoint. These properties alone do not guarantee that the error is fixed. [M2]

Concept only; not ready-to-use AL code:

```
Parent API Page: PageType = API
                  ODataKeyFields = SystemId
                  DelayedInsert = true
Part items:      SubPageLink = Parent SystemId → Parent.SystemId
Child:           valid parent reference before persistence
```

Option B - Dedicated receiving layer. The ERP partner provides a receiver that accepts the WAMAS contract and creates BC records internally. If the BC OData interface cannot directly provide the required request/response format, the partner must propose a technically supported service or adapter approach. A standard AL API page must not be assumed to support an arbitrary REST body and arbitrary JSON response. An additional component would be an architectural change requiring explicit agreement.

Transactions and error responses

TARGET: Before implementation, agree whether an entire batch is rolled back or individual items can be accepted. The NAS receiver operates per item; its error example is `{ "items": [ { "id": 505, "msg": "Message could not be stored: ..." } ] }`. BC may adopt this format only after checking the applicable WAMAS DI API version. For partial failures, sender and receiver must know which IDs to resend. Technical errors must not be returned as an empty success list.

TARGET: Retries must produce the same business effect at most once. Agree a stable key combining sender context and WAMAS id or telegram transaction; account for company, restarts and ID reuse. Durable acceptance, business processing and business errors must be individually traceable.

8 Handover and acceptance criteria

Responsibility	Required deliverable
ERP/BC partner	Corrected extension or receiver; exact URL; request/response contract; BC app permissions; parent/child evidence; error and duplicate handling; telemetry analysis.
SSI/WAMAS	Version and DI API specification; OAuth/OIDC configuration including JWK; sender rules for status code and body; retry, timeout and batch size; transmission evidence.
Joint responsibility	Agree test data and expected business results; execute the tests below; retain raw responses without tokens; document production approval.

BC access: Enable the application for the target company and assign the necessary API, table and codeunit permissions. Microsoft describes Entra application permissions and the additional app access configuration in BC; the partner must confirm the required object permissions. Successful test access does not replace documented operational configuration. [M1]

No.	Acceptance test	Pass condition
A01	OAuth2 directly from WAMAS	Token acquisition and renewal work; correct tenant/scope; JWK configuration confirmed.
A02	Native requests 505, 516 and 517	All three complete telegram types are accepted without a preliminary parent POST; agreed 2xx code and valid response body.
A03	Inspect parent and items	Valid parent persisted; all items correctly assigned; data unchanged; no dependency on Entry No.=0.
A04	Actual WAMAS sender	WAMAS interprets the same response as successful transmission; no unjustified retry; BC message can be located.
A05	Business processing	For all three types, order/item, quantities, direction and status match the expected results agreed beforehand.
A06	Multiple items and partial failure	Agreed atomicity is maintained; errors identify the affected ID; retries do not create duplicate postings.
A07	Timeout after persistence and duplicate	Retry is reliably recognised; the same message does not produce a second business effect.
A08	Missing company and invalid token	Defined error response; no storage in the wrong company; no empty success response on failure.
A09	Invalid data and unknown converter	Controlled error rather than data loss; WAMAS retry/error state follows the agreed contract.
A10	Operation and load limits	Maximum payload/batch size, timeout, 429/5xx handling, monitoring and restart behaviour documented and tested.

Acceptance evidence for each test: environment and extension version, WAMAS version, date/time with time zone, request file and hash, WAMAS id/transaction ID, HTTP code, sanitised response, BC parent/child identities and result in the WAMAS sender. Acceptance requires joint confirmation; this document is not production approval.

9 Sources and evidence register

Evidence is referenced by ID in the text and test matrix. Screenshots are embedded unchanged. The selected images contain no visible client secrets, passwords or bearer tokens. Images containing credentials are excluded.

ID	Source and scope of evidence
G	Conversation “REWST API Verbindung testen”, 23 September 2026. ID 6ab38ace-2fd4-83eb-98e0-0781aafbfbf7. Chronology, token/minimal POST results and metadata description. Company list provided verbatim by the user.
Q1	main.py, FastAPI receiver version 3.3.0. Models, DI POST route, archive mapping, Basic Auth, storage and response logic directly inspected.
Q2	Pasted markdown.md, NAS export with 35 records. Selected archive IDs 4/15/16 with WAMAS IDs 505/516/517. Complete data strings directly inspected.
B1	Original screenshot: GET without company, 404 Internal_CompanyNotFound. Appendix A.
B2	Original screenshot: verification GET with company and expand, 200 OK, entryNo 2 and 1 with empty items. Appendix B.
B3	Original screenshot: source plus items, abbreviated WLSSDACK002, 404 Internal_RecordNotFound. Appendix C.
B4	Original screenshot: native items format with WAMAS id 505, 404 Internal_RecordNotFound. Appendix D.

Evidence classification

Token 200, initial 401 responses, minimal POST 201 and the details of \$metadata are classified here as conversation evidence. The early original images and original metadata XML are not included in the embedded evidence set. For auditable acceptance, retain sanitised original responses and the current metadata XML. The Postman screenshots show tests against BC; they do not replace end-to-end evidence from WAMAS.

Vendor references for implementation

M1 - Microsoft Learn, Using service-to-service (S2S) authentication. Reference for client credentials, scope, token URL and app access in BC. Accessed 23 September 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. Reference for deep insert and modelling with SystemId and SubPageLink. Accessed 23 September 2026.

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

The vendor references describe technical capabilities; they do not prove the implementation of the custom ACU extension. SSI must provide the applicable WAMAS DI API specification for the deployed version.

Reproducible payloads in the test package

The three request_*.json files reconstruct the outer batch from the archive. Only payload.wamas_message_id is mapped back to items[id]. The content of converter, data and processingPath is unchanged. These are not recorded HTTP captures. The files contain no authentication data. A manifest provides SHA-256 checksums.

Appendix A A company must be specified

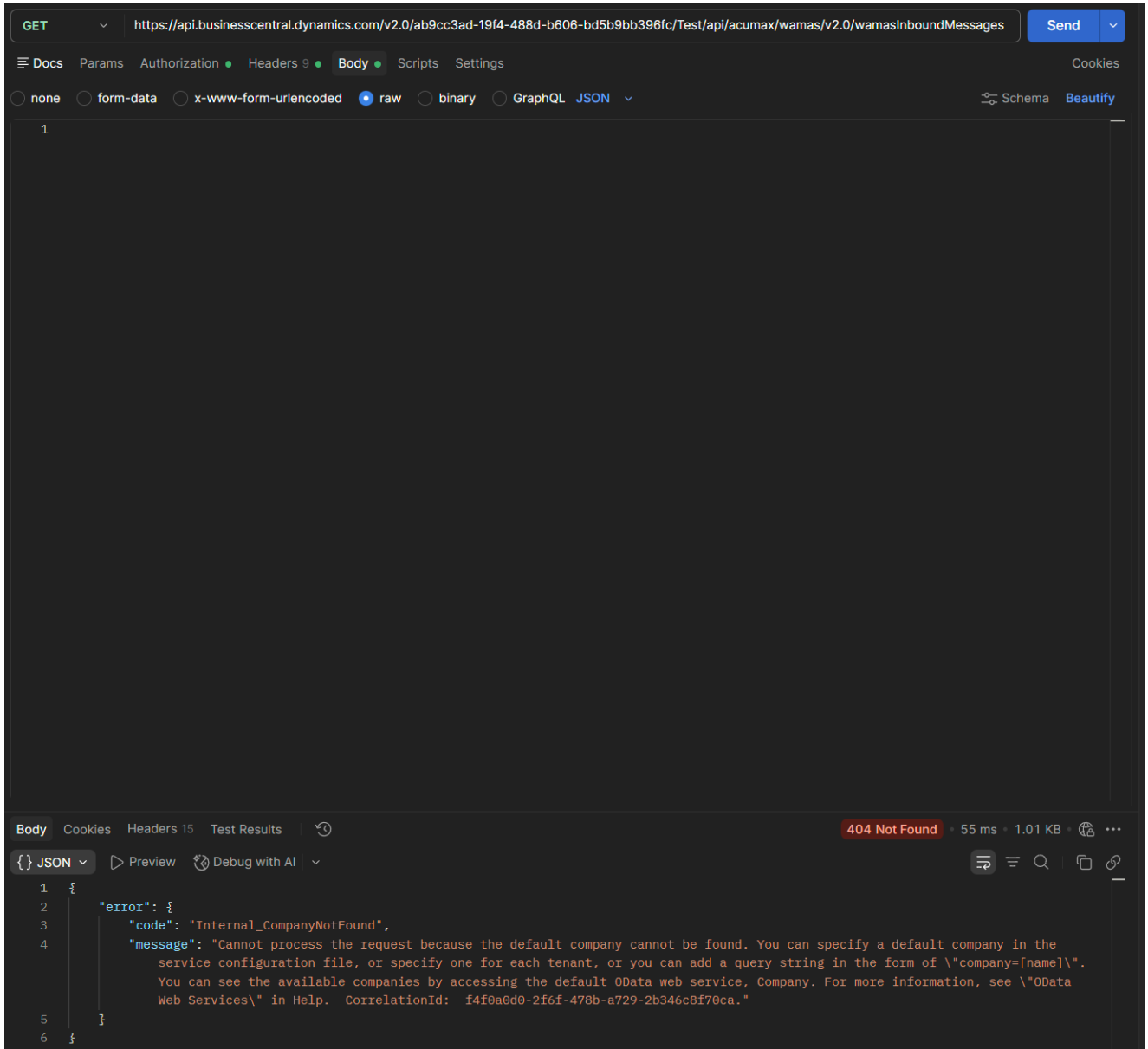


Figure 2 - B1. GET without company returning 404 Internal_CompanyNotFound. The response requests company selection. This screenshot evidences the later repeated company error, not the initial 401.

Appendix B Reading with a company succeeds

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/wamasInboundMessages`
- Method:** GET
- Query Params:**

Key	Value	Description
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/\"JzE5Ozg2Njk5MTYwOTAyNDQzNzYxNDcxOzAwOyc=\"",
14      "id": "6b7dd6d8-ccb5-f111-aaa8-6045bd2c3231",
15      "entryNo": 1,
16      "status": "Message received",
17      "source": "REST",
18      "items": []
19    }
20  ]
21 }

```

Figure 3 - B2. Verification GET with company=AcuMax Med AG, ordering, top and expand. HTTP 200; entryNo 2 and 1 are visible, each with an empty items array.

Appendix C The first deep insert fails

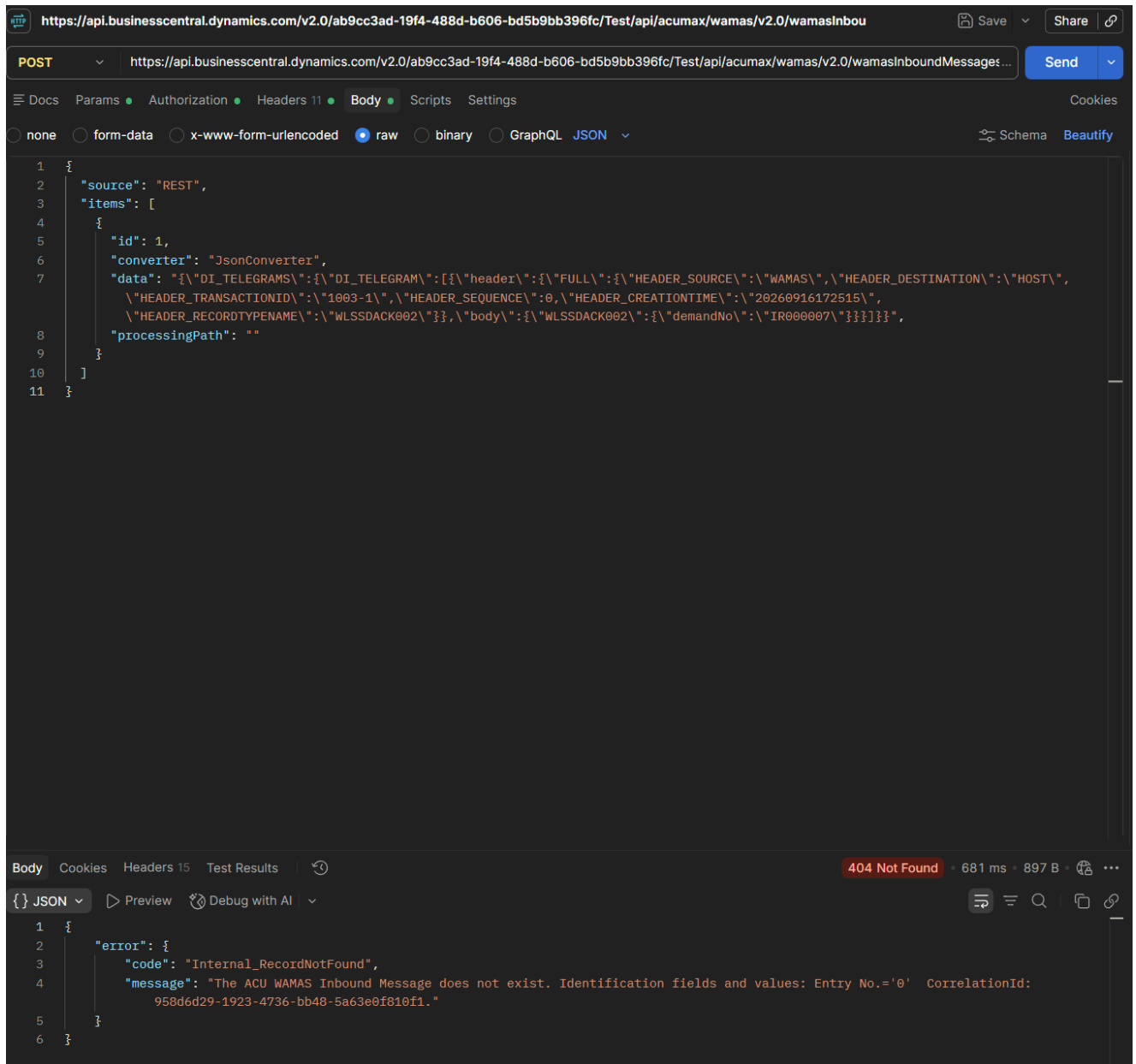


Figure 4 - B3. POST with source=REST and an abbreviated WLSSDACK002. HTTP 404 Internal_RecordNotFound, Entry No.=0. CorrelationId: 958d6d29-1923-4736-bb48-5a63e0f810f1.

