

EXTERNAL CONNECTIONS - APPLICATIONS

Open Entity facilitates external connections to Business Central through a single API for entering and modifying any type of information, such as a new customer or contact, a sales document, an inventory, etc. The API **'entityEntries'** will always be used for this purpose.

For other operations, such as reading or deleting an entity or retrieving a serial number from Business Central, we use the **OE API** followed by an underscore and the desired function, such as **ReadEntity** or **DeleteEntity** (e.g., **OE_ReadEntity**).

All information sent or received must be compressed in **GZIP** format. Both in sending and receiving, it is necessary to specify the template that will manage the information and a user or device identifier. This is more of a disabling mechanism for handling any potential issues or for planned inactivity of the device or user, rather than a security measure.

In this document, we will use the **Postman** application to simulate an application that sends and receives information to and from Business Central, and the **DevToys** program to compress/decompress information in GZIP format.

DATA READING

For both reading and writing, the action to be performed will always be the **POST method**, and OAuth (token) authorization is essential to establish the connection. In Postman, we used the **Bearer** option and the token previously read from the Open Entity application.

The screenshot shows a Postman HTTP client interface. The top bar indicates the request method is **POST** and the URL is `https://login.microsoftonline.com/{tenant-id}/oauth2/v2/token`. The **Body** tab is selected, showing a JSON object with the following fields:

Key	Type	Value	Description
☒ grant_type	Text	client_credentials	
☒ client_id	Text		
☒ client_secret	Text		
☒ scope	Text	https://api.businesscentral.dynamics.com/.default	

The bottom bar shows the response status is **200 OK**, time is **267 ms**, and size is **2.19 KB**. The response body is a JSON object with the following fields:

```
1 { "token_type": "Bearer", "expires_in": 3599, "ext_expires_in": 3599, "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6IjVCM25SeHRRN2pp0GVORGMzRnkWNUtmOTdaRSIsImtpZCI6IjVCM25SeHRRN2" }
```

The parameters **client_id** and **client_secret** correspond to the values of our application registration in **Azure Microsoft Entra**.

The **access_token** received will be included in our data queries or insertions in the **Authorization** tab. For example, **Bearer + received token**.

To perform a data read, we will need to send the following information:

- **templateCode**: the template code that will read the data.
- **filterTable**: filters to be used in data selection.
- **senderUser**: a name that must exist in **Permissions** within the Open Entity application for the template used.
- **format**: the format in which we want the information returned; it can only be **xml** or **json**.
- **skipNameSpace**: Open Entity uses namespaces in tables for self-management; here, we can decide whether we want it included in the data received.

In the following request, we are requesting information about a customer with a specific Vat Number.

The screenshot displays a REST client interface for a POST request to the Business Central API. The URL is `https://api.businesscentral.dynamics.com/v2.0/[TENANT GUID]/SandboxDev2/ODataV4/OE_ReadEntity?company=8586dd27-55e9-ed11-884e-6045bdc8c698`. The request body is a JSON object with the following fields:

```
1 {
2   "templateCode": "SCENARIO_04--CUST",
3   "filterTable": "cust#vat;B16648305",
4   "senderUser": "Pos001",
5   "format": "xml",
6   "skipNameSpace": true
7 }
```

The response body is a JSON object with the following fields:

```
1 {
2   "@odata.context": "https://api.businesscentral.dynamics.com/v2.0/[TENANT GUID]/SandboxDev2/ODataV4/$metadata#Edm.String",
3   "value": "H4sIAAAAAAAC12PTQuCQBCG0wF9h2XPth+Y5WFdQaGTe0iD6DjoEoLugJr081s3g2g0M++8PAzvqPTVtwQy/dCgTahkghJjK6wb+0jo9XLcxpQMI9gaWrQmoRZpqjdrdUIc3SREVC/ho5wuke1IuFJ81l8X0qNL7KCG6/QmIGd"
4 }
```

If we decompress the value received in “**value**”, we will see that the read action was executed successfully and the data corresponds to the configuration of the template defined in our sent **templateCode**:

Comprimir/Descomprimir usando GZip

Configuración

↔ Comprimir/Descomprimir usando GZip
Seleccione comprimir o descomprimir

Descomprimir ☐

Entrada

Pegar ✕

H4sIAAAAAAACL2PTQuCQBCG0wf9h2XPth+Y5WfdQaGTe0iD6DjoEoLugJr081s3g2g0M++8PAzvqPTVtWQy/dCgTahkghJjk6wb+
0jo9XLcxpQMI9gaWrQmoRZpqjdrdUIc3SREvc/ho5wukeLIuFJ8l18X0qNL7KCGG/QmIGdWM0KQ2V+YvDd1MxZN51p+1zIK/JU/e4EzaMFwXoMikOGBhbHiP+7CTTDqT073uzgUkeLz6iPzJbPi/o3VG7mDAr0GAQAA

Salida

↗ Copiar

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Root>
  <cust>
    <No.>50000</No.>
    <Name>NomadaWare, S.L. </Name>
    <CreditLimitLCY>15,000</CreditLimitLCY>
    <BalanceLCY>10,137.38</BalanceLCY>
    <vat>B16648305</vat>
  </cust>
</Root>
```

Let's look at the configuration of the "SCENARIO 04 - CUST" template:

Template User

+

✓ Saved

SCENARIO 04 - CUST

Copy-Paste

Export Data

Excel

Run Process

Save File

Archive

Copy Template

Permissions

Vars & Formulas

Get All Fields & Enable

Delete Obsolete/Unexistent Fields

Get Tables Related

...

General

CodeSCENARIO 04 - CUST

Source Table ID18

Source NameCustomer

DescriptionGet the customer code, name and credit for an app.

OperationOnly Read

Table Trigger

Permissions0

Allow Requests

Enable

Use-Make Equivalence

FolderSCENARIO 04

Template Use0

Lines

New LineDelete LineUse KeyClear KeyExcel HeaderExcel LineIndent

Target Table ID	Target Table Name	Indentation	Related Table ID	Tag	Operation	Table Trig...	Key	Fields	Formu...	Filters	Fusion	Excel	Ena...	Force (Imp...
→ 18	Customer	0	18	cust	Only Read	☐		5	No	0	No	0	☒	☐

TableData - 18

✓ Saved

Field Lines

Search

Analyze

New

Edit List

Delete

Delete All

Get & Enable All

Enable All

Disable All

Check Duplications

Enable Selected

Disable Selected

Check formulas

...

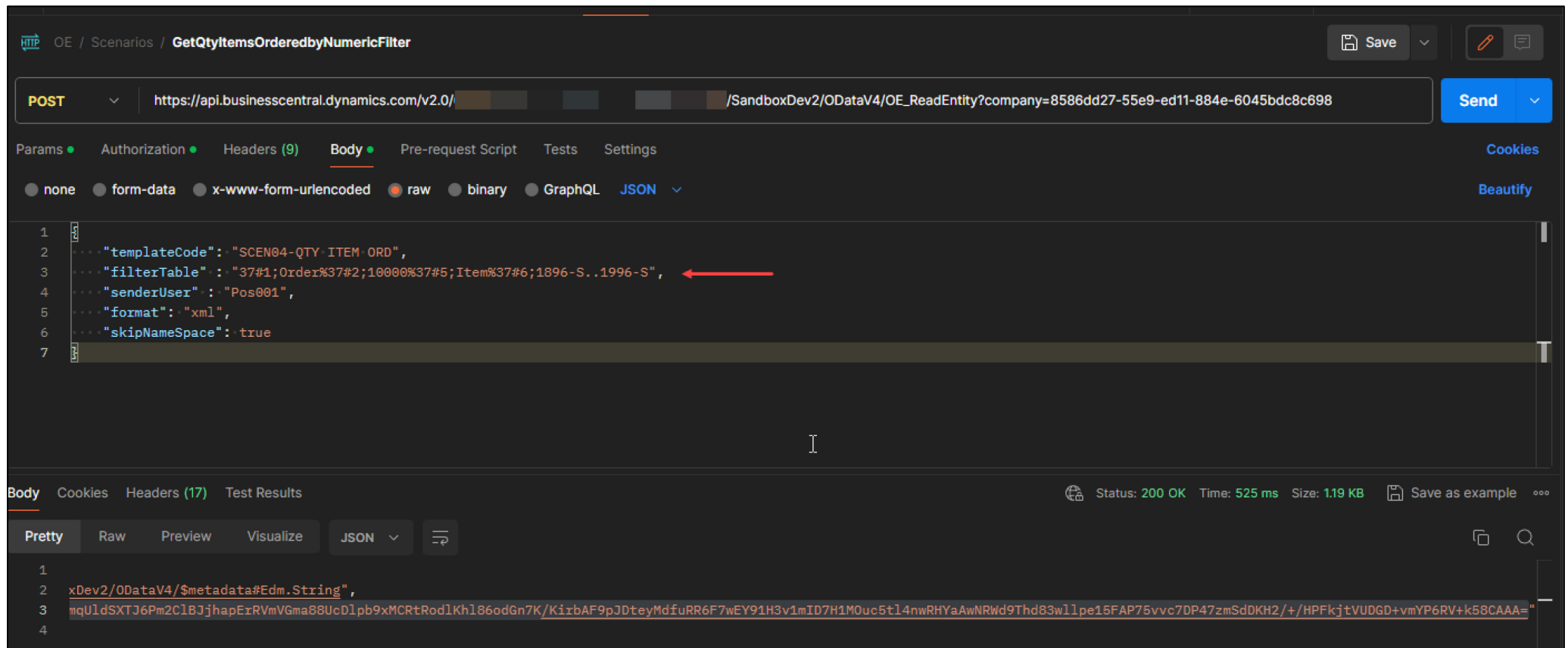
Field ID	Field Name	Tag	Order ↑	Serie No.	Get From	Field Value	Mapping (Imports)	Ena... Map	Variable Id	Formula	Con...	Ign... For...	Equal Field No.	Fuse	Cha... Sign (nu...	Vali...	Edit...	Visi...	Ena...
→ 1	No.		0				0	☐			No	☐	0		☐	☐	☐	☐	☒
2	Name		0				0	☐			No	☐	0		☐	☐	☐	☐	☒
20	Credit Limit (LCY)		0				0	☐			No	☐	0		☐	☐	☐	☐	☒
59	Balance (LCY)		0				0	☐			No	☐	0		☐	☐	☐	☐	☒
86	VAT Registration No.	vat	0				0	☐			No	☐	0		☐	☐	☐	☐	☒

As can be observed, the operation of the template is **read-only**, and the tags that make up the selection filter must match. This is all we need for our data request to work.

FILTERS

There are two ways to filter information from an external app: by using the tags defined in the template lines and fields or simply by their numbering. Combining both syntaxes in a single filter is not allowed.

Both filter types consist of three main values separated by special characters. The values are the table number or name (tag), the field number or name (tag), and the filter itself or selection values. Because it is possible to filter by any level of the template, each individual filter must always contain these three values.



The screenshot shows a REST client interface with a POST request to the endpoint `https://api.businesscentral.dynamics.com/v2.0/[redacted]/SandboxDev2/ODataV4/OE_ReadEntity?company=8586dd27-55e9-ed11-884e-6045bdc8c698`. The request body is a JSON object with the following fields:

```
1 {
2   "templateCode": "SCEN04-QTY-ITEM-ORD",
3   "filterTable": "37#1;Order%37#2;10000%37#5;Item%37#6;1896-S..1996-S",
4   "senderUser": "Pos001",
5   "format": "xml",
6   "skipNameSpace": true
7 }
```

A red arrow points to the `filterTable` field value. The response status is 200 OK, with a time of 525 ms and a size of 1.19 KB. The response body is a long string of XML data.

In this numeric filter, **37** is the **Sales Line** table; the numbers **1**, **2**, **5**, and **6** are the fields, and **Order**, **10000**, **Item**, and **1896-S..1996-S** are the values to filter.

The separators have the following meanings:

- The ‘;’ character separates the field from the filter value → **6;1896-S..1996-S** (6 is the field No., and 1896-S..1996-S is the product range).
- The ‘#’ character separates the table from the previous filter (field and selection). **37#6;1896-S..1996-S** means that for table 37 (Sales Line), the described filter is applied.
- Finally, the ‘%’ character separates different filter groups that may be configured.

In this case:

37#1;Order

37#2;10000

37#5;Item

37#6; 1896-S..1996-S

The following breakdown means to select all sales lines of an Order document for customer 10000 where the lines are of type Item and the product codes fall within the range from 1896-S to 1996-S. Filters with alphanumeric encoding (names) must match the tag configured in the template fields.

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

- Method:** POST
- URL:** `https://api.businesscentral.dynamics.com/v2.0/[redacted]/SandboxDev2/ODataV4/OE_ReadEntity?company=8586dd27-55e9-ed11-884e-6045bdc8c698`
- Body Type:** JSON
- Body Content:**

```
1 {
2   "templateCode": "SCEN04-QTY-ITEM-ORD",
3   "filterTable": "salesLine#doctype;Order%salesLine#custcode;10000%salesLine#type;Item%salesLine#itemcode;1896-S..1996-S",
4   "senderUser": "Pos001",
5   "format": "xml",
6   "skipNameSpace": true
7 }
```
- Status:** 200 OK, Time: 638 ms, Size: 1.19 KB
- Response Body:**

```
1
2 "@odata.context": "https://api.businesscentral.dynamics.com/v2.0/[redacted]/SandboxDev2/ODataV4/$metadata#Edm.String",
3 "value": "H4sIAAAAAAACs2SywzCMBBFXQv+Q8i+pu2iVpim00EQiq8PC00ggTaRjhX794ZYi4KLrsRswi3z4B4Yy09NTW7YGq1VRqN1SAmqUldSXTJ6Pm2C1BjhapErRVmVGma88UcD1pb9xMCRtRod1Kh186odGn7K/KirbAF9pJDteyM"
```

Template User ✓ Saved

SCEN04-QTY ITEM ORD

Copy-Paste Export Data Excel Run Process Save File Archive Copy Template Permissions Vars & Formulas Get All Fields & Enable Delete Obsolete/Unexistent Fields Get Tables Related ...

General

Code: SCEN04-QTY ITEM ORD

Source Table ID: 18

Source Name: Customer

Description: Returns the Outstanding Quantity of Items

Operation: Only Read

Table Trigger: ☐

Permissions: 2

Allow Requests: ☐

Enable: ☒

Use-Make Equivalence: ☐

Folder: SCENARIO 04

Template Use: 0

Lines

New Line Delete Line Use Key Clear Key Excel Header Excel Line Indent

Target Table ID	Target Table Name	Indentation	Related Table ID	Tag	Operation	Table Trig...	Key	Fields	Formu...	Filters	Fusion	Excel	Ena...	Force (Imp...
→ 37	Sales Line	0	18	salesLine	Only Read	☐		5	No	0	Yes	0	☒	☐

TableData - 37 ✓ Saved

OE Field Lines ✓ 🔍 📄 + New 🔧 Edit List 🗑️ Delete 🗑️ Delete All 🔑 Get & Enabled All 🔑 Enabled All 🔑 Disable All 🔍 Check Duplicates 🔑 Enabled Selected 🔑 Disable Selected 📊 Check formulas ⋮ More options 📄

Field ID	Field Name	Tag	Order	Series No.	Get Series When	Get From	Field Value	Mapping (Imports)	Ena... Map	Variable Id	Formula	Condi...	Ign... For...	Equal Field No.	Fuse	Cha... Sign (nu...	Vali...	Edit...	Visi...	Ena...	Lookup Page No.	Lookup Field No.	Field Type
→ 1	Document Type	doctype	10					0	☐			No	☐	0		☐	☐	☐	☐	☒	0	0	Option
2	Sell-to Customer No.	custcode	20					0	☐			No	☐	0		☐	☐	☐	☐	☒	0	0	Code
5	Type	type	30					0	☐			No	☐	0		☐	☐	☐	☐	☒	0	0	Option
6	No.	itemcode	50					0	☐			No	☐	0	Group	☐	☐	☐	☐	☒	0	0	Code
16	Outstanding Quantity		60					0	☐			No	☐	0	Total	☐	☐	☐	☐	☒	0	0	Decimal

ALPHANUMERIC FILTERS

salesLine#doctype;Order

salesLine#custcode;10000

salesLine#type;Item

salesLine#itemcode; 1896-S..1996-S

Highlight the use of the **Fuse** column to group product codes and total them in a single line.

RESULT

Comprimir/Descomprimir usando GZip

Salida

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Root>
  <salesLine>
    <doctype>Order</doctype>
    <custcode>10000</custcode>
    <type>Item</type>
    <itemcode>1996-S</itemcode>
    <OutstandingQuantity>12</OutstandingQuantity>
  </salesLine>
  <salesLine>
    <doctype>Order</doctype>
    <custcode>10000</custcode>
    <type>Item</type>
    <itemcode>1968-S</itemcode>
    <OutstandingQuantity>10</OutstandingQuantity>
  </salesLine>
  <salesLine>
    <doctype>Order</doctype>
    <custcode>10000</custcode>
    <type>Item</type>
    <itemcode>1928-S</itemcode>
    <OutstandingQuantity>7</OutstandingQuantity>
  </salesLine>
</Root>
```

INSERTING-MODIFYING DATA

An external application generates the following data in XML (it also supports JSON) to create an order and compresses it in GZIP format:

Comprimir/Descomprimir usando GZip

Configuración

Comprimir/Descomprimir usando GZip
Seleccione comprimir o descomprimir

Entrada

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Root>
  <cust>
    <VATRegistrationNo.>B16648305</VATRegistrationNo.>
  </cust>
  <salesLine>
    <No.>1896-S</No.>
    <Quantity>10</Quantity>
  </salesLine>
  <salesLine>
    <No.>1900-S</No.>
    <Quantity>5</Quantity>
  </salesLine>
  <salesLine>
    <No.>1936-S</No.>
    <Quantity>25</Quantity>
  </salesLine>
  <salesLine>
    <No.>1996-S</No.>
  </salesLine>
</Root>
```

Salida

```
H4sIAAAAAAAC53R3QqCMBQA4CfwHcbudZuWKMxJXXQVQWbdDx0ysA3cjHr7pLgQaFB3Z+fn4xxG8/u1BTfRGa1VBkmAIRCq0rVUTQbP5c5PIDCWq5q3WokMKg1z5tFCa8s8AGjVmzFw4WTFqKRxncOu2gA7Y1cbxKIrymaKY6zKMXQA1vhdLLJSZu6CBJGvsnigZulz32XF1pH4xgit4Pb5A+gHkuxXiBcxv+rkVLy4V/cYu3ht9upWj8jCd+omsgygEAAA==
```

salesorderexample.xml

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <Root>
3   <cust>
4     <VATRegistrationNo.>B16648305</VATRegistrationNo.>
5   </cust>
6   <salesLine>
7     <No.>1896-S</No.>
8     <Quantity>10</Quantity>
9   </salesLine>
10  <salesLine>
11    <No.>1900-S</No.>
12    <Quantity>5</Quantity>
13  </salesLine>
14  <salesLine>
15    <No.>1936-S</No.>
16    <Quantity>25</Quantity>
17  </salesLine>
18  <salesLine>
19    <No.>1996-S</No.>
20    <Quantity>20</Quantity>
21  </salesLine>
22 </Root>
```

salesorderexample.json

```
1 {
2   "cust": {
3     "VATRegistrationNo.": "B16648305"
4   },
5   "salesLine": [
6     {
7       "No.": "1896-S",
8       "Quantity": "10"
9     },
10    {
11      "No.": "1936-S",
12      "Quantity": "25"
13    },
14    {
15      "No.": "1996-S",
16      "Quantity": "20"
17    }
18  ]
19 }
```

We copy the Compressed Output and add it to the **entity** field in our Post, enclosed in quotes. The compressed data can be in XML or JSON format.

The general data entry URL is as shown on the screen below. The blank space corresponds to our **tenant id**, and the underlined text should be our **company id**. The final text, **entityEntries**, is the name of our resource or API.

The screenshot displays a REST client interface with a POST request to the URL: `https://api.businesscentral.dynamics.com/v2.0/[Tenant id]/SandboxDev2/api/nomadaware/oe/v2.0/companies(8586dd27-55e9-ed11-884e-6045bdc8c698)/entityEntries`. The request body is a JSON object with the following fields: `templateCode` (SCENARIO 04 - ORDER), `senderUser` (Pos001), and `entity` (a long alphanumeric string). The response status is 201 Created, and the response body is a JSON object with the following fields: `@odata.context`, `@odata.etag`, `trackCode` (1ca50582-6c8c-4ef3-9598-0274fcdc2cef), `templateCode` (SCENARIO 04 - ORDER), `senderUser` (Pos001), `operation` (Insert-Modify), and `entity` (a long alphanumeric string). A red arrow points to the `trackCode` field in the response body.

```
POST https://api.businesscentral.dynamics.com/v2.0/[Tenant id]/SandboxDev2/api/nomadaware/oe/v2.0/companies(8586dd27-55e9-ed11-884e-6045bdc8c698)/entityEntries

{
  "templateCode": "SCENARIO 04 - ORDER",
  "senderUser": "Pos001",
  "entity": "H4sIAAAAAAACB3R3QqCMBQA4CfwhCbudZuWKMxJXXQVQWbdDx0ysA3cjHr7p1gQaFB3Z+fn4xxG8/u1BTfRGalVBkmAIRCq0rVUTQbP5c5PIDCWq5q3WokMKg1z5tFCa8s8AGjVmzFw4WVTFqKRxbnc0u2gA7Y1cbxKIrymaKY6zKMxQA1vhd1LJSZu6CBJGvsniqZulz32XF1pH4xgit4Pb5A+gHkuxXiBcxv+rkVLy4V/cYu3ht9upWj8jCd+omsgygEAAA=="
}
```

```
{
  "@odata.context": "https://api.businesscentral.dynamics.com/v2.0/[Tenant id]/SandboxDev2/api/nomadaware/oe/v2.0/$metadata#companies(8586dd27-55e9-ed11-884e-6045bdc8c698)",
  "@odata.etag": "W/\"JzE50zgZODU3NzY3NDU2NjQyMjU3MTkxOzAwOyc=\"",
  "trackCode": "1ca50582-6c8c-4ef3-9598-0274fcdc2cef",
  "templateCode": "SCENARIO 04 - ORDER",
  "senderUser": "Pos001",
  "operation": "Insert-Modify",
  "entity": "H4sIAAAAAAACrWSXQuCMBSGuw76D2P3u1kpK10pi64i6PN+1ZCBnYFbUf++KWQGCRF4dTjvDs/D0Yx190uBbqLUUkGCPZdiJ0CkzhLyB093CyfESBs0Z14oEakGhbN0NGQbpYytCDHDj4WIT1dtkGWBjusgwVwC6xE7dJjtNikX2pTcwNnkue"
}
```

The returned value of the **'trackCode'** field is the tracking code for the operations performed by our Post in Business Central.

The result is the following order created in Business Central:

Sales Order

✓ Saved

101292 · NomadaWare, S.L.

Home

Prepare

Print/Send

Request Approval

Order

More options

Post...

Release

Create Warehouse Shipment

Create Inventory Put-away/Pick...

Archive Document

General

Show more

Customer Name

NomadaWare, S.L.

...

Order Date

1/30/2024

Contact

Diego Echevarria

...

Due Date

2/13/2024

Document Date

1/30/2024

Requested Delivery Date

Posting Date

1/30/2024

External Document No.

VAT Date

1/30/2024

Status

Open

Lines

Manage

Line

Order

New Line

Delete Line

Select items...

Type	No.	Item Reference No.	Description	Loc... Code	Quantity	Qty. to Assem... to Order	Res... Qua...	Unit of Measure Code	Unit Price Excl. VAT	Line Discount %	Line Amour Excl. VA
→ Item	:	1896-S	Escritorio ATENAS NEW NEW		10		—	UDS	25.00		250.0
Item		1900-S	Silla inv. PARÍS, negra		5		—	UDS	90.00		450.0
Item		1936-S	Silla invit. BERLÍN, amar.		25		—	UDS	150.00		3,750.0
Item		1996-S	Pizarra ATLANTA, base		20		—	UDS	55.00		1,100.0

Details

Attachments (0)

Cartera Receiv. Statistics FB

No. of Bills

Open Bills0

Open Bills in Bill Gr.0

Open Bills in Post. Bill Gr.0

Hon. Bills in Post. Bill Gr.0

Rej. Bills in Post. Bill Gr.0

Redr. Bills in Post. Bill Gr.0

Hon. Closed Bills0

Rej. Closed Bills0

Remaining Amt. (LCY)

Open Bills0.00

Open Bills in Bill Gr.0.00

Open Bills in Post. Bill Gr.0.00

Hon. Bills in Post. Bill Gr.0.00

Rej. Bills in Post. Bill Gr.0.00

Redr. Bills in Post. Bill Gr.0.00

Hon. Closed Bills0.00

Rej. Closed Bills0.00

Cartera Fact. Statistics FB

TEMPLATE SCENARIO 04 - ORDER (Business Central)

This is the composition in Open Entity for the previous example of creating an order. It uses resources covered in other documents of the manual, so we will simply highlight them graphically.

Template Lines:

SCENARIO 04 - ORDER

Copy-Paste

Export Data

Excel

Run Process

Save File

Archive

Copy Template

Permissions

Vars & Formulas

Get All Fields & Enable

Delete Obsolete/Unexistent Fields

Get Tables Related

...

General

Code

SCENARIO 04 - ORDER

Permissions

2

Source Table ID

18

Allow Requests

Source Name

Customer

Enable

Description

Import Sales Order

Use-Make Equivalence

Operation

Insert-Modify

Folder

SCENARIO 04

Table Trigger

Template Use

0

Lines

New Line

Delete Line

Use Key

Clear Key

Excel Header

Excel Line

Indent

	Target Table ID	Target Table Name	Indentation	Related Table ID	Tag	Operation	Table Trig...	Key	Fields	Formu...	Filters	Fusion	Excel	Ena...	Force (Imp...
→	18	Customer	0	18	cust	Only Read			2	No	0	No	0		
	36	Sales Header	2	18		Insert-Mod...			3	Yes	0	No	0		
	37	Sales Line	4	36	salesLine	Insert-Mod...			7	Yes	0	No	0		

Table Fields Configuration:

TableData - 18

Field Lines

Field ID

Field Name

Tag

Order ↑

Series No.

Get From

Field Value

Mapping (Imports)

Ena... Map

Variable Id

Formula

Con...

Ign... For...

Equal Field No.

Fuse

Cha... Sign (nu...

Vali...

Edit...

Visi...

Ena...

→	86	VAT Registration No.		10				0	☐	vatRegistration		No	☐	0		☐	☐	☐	☐	☒
	1	No.		20		GETCUSTBYVAT		0	☐	custNo		No	☐	0		☐	☐	☐	☐	☒

TableData - 36

Field Lines

Field ID

Field Name

Tag

Order ↑

Series No.

Get From

Field Value

Mapping (Imports)

Ena... Map

Variable Id

Formula

Con...

Ign... For...

Equal Field No.

Fuse

Cha... Sign (nu...

Vali...

Edit...

Visi...

Ena...

→	1	Document Type		10			Order	0	☐			No	☐	0		☐	☐	☐	☐	☒
	3	No.		20		V-PED		0	☐	serieNo		No	☐	0		☐	☐	☐	☐	☒
	2	Sell-to Customer No.		40				0	☐		custNo	No	☐	0		☐	☒	☐	☐	☒

TableData - 37

Field Lines

Field ID

Field Name

Tag

Order ↑

Series No.

Get From

Field Value

Mapping (Imports)

Ena... Map

Variable Id

Formula

Con...

Ign... For...

Equal Field No.

Fuse

Cha... Sign (nu...

Vali...

Edit...

Visi...

Ena...

→	1	Document Type		0			Order	0	☐			No	☐	0		☐	☐	☐	☐	☒
	2	Sell-to Customer No.		0				0	☐		custNo	No	☐	0		☐	☐	☐	☐	☒
	3	Document No.		0				0	☐		serieNo	No	☐	0		☐	☐	☐	☐	☒
	4	Line No.		0				0	☐	line	line + 10000	No	☐	0		☐	☐	☐	☐	☒
	5	Type		0			Item	0	☐			No	☐	0		☐	☐	☐	☐	☒
	6	No.		0				0	☐			No	☐	0		☒	☐	☐	☐	☒
	15	Quantity		0				0	☐			No	☐	0		☒	☐	☐	☐	☒

'Get From' Card Configuration used in the field configuration of **Customers** (Get From column):

Get From Header

✓ Saved

GETCUSTBYVAT

General

Code

GETCUSTBYVAT

Get From Return Field

Source Table ID

18

Field ID

1

Source Name

Customer

Name

No.

Description

Get the customer code searching

Enable

☒

Suitable for

Get From

GetFrom Use

3

Lines

New Line

Delete Line

Field ID	Field Name	Operator	Field Value	Is a Var	Order ↑	Ena...
86	VAT Registration No.	=	vatRegistration	☒	10	☒

To modify a single order line, we can use the same request but with a different template as the sending parameter (**SCEN04-READORDRLINES**).

POST

https://api.businesscentral.dynamics.com/v2.0/3/OpenEntity_Demo/api/nomadaware/oe/v2.0/companies(3967b449-5ae6-ee11-a201-6045bdacc8c3)/entityEntries

Send

Params

Authorization

Headers (10)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "templateCode" : "SCEN04-READORDRLINES",
3   "senderUser" : "Pos001",
4   "entity": "H4sIAAAAAAAC0102wrCMAz9gv1D6ftsByoKWfcm+CR4+YCxBilsia5F3N/bdk5MHnJODjk50LyHXrxw9I6pltVKS4HUsXV0r
5     +Xteih3UvjQkm17JqwlswxMAWfmYAohwPe00CcYieUuTA80p9HiCGqhP5HYVDx2JmuRzUqyMGsdC1TG8zrFHgM0oP5sXFwkn73e1hdQXzprzzDFB6DSTOHUkg5UDvwB+uffDe4AAAA="
6 }
```

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <Root>
3   <slines>
4     <doctype> Order</doctype>
5     <docno> 101015</docno>
6     <line> 40000</line>
7     <type> Item</type>
8     <itemno> 1906-S</itemno>
9     <qty> 10</qty>
10    </slines>
11  </Root>
```

DELETING DATA

To delete data related to a template, we use the '**OE_DeleteEntity**' function.

The expected parameters are the template code, filters, and the user or device that wants to execute the action.

In this case, we are attempting to delete the second order line we just created. However, Business Central prevents this and notifies us that the user **Pos001** does not have permission to perform the action.

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

- URL:** `https://api.businesscentral.dynamics.com/v2.0/t.../SandboxDev2/ODataV4/OE_DeleteEntity?company=8586dd27-55e9-ed11-884e-6045bdc8c698`
- Method:** POST
- Body (JSON):**

```
1 {
2   "templateCode": "SCEN04-READORDLINES",
3   "filterTable": "slines#doctype;Order%slines#docno;101292%slines#line;20000",
4   "senderUser": "Pos001"
5 }
```
- Status:** 200 OK (Note: The status is 200 OK, but the response body indicates an error).
- Time:** 413 ms
- Size:** 988 B
- Response Body (JSON):**

```
1 {
2   "@odata.context": "https://api.businesscentral.dynamics.com/v2.0/t.../SandboxDev2/ODataV4/$metadata#Edm.String",
3   "value": "The user hasn't permission for that template!"
4 }
```

The template in Business Central that can help us modify or delete an order line is as follows:

Template User

SCEN04-READORDLINES

Copy-PasteExport DataExcelRun ProcessSave FileArchiveCopy TemplatePermissionsVars & FormulasGet All Fields & EnableDelete Obsolete/Unexistent FieldsGet Tables Related

General

CodeSCEN04-READORDLINES

Source Table ID37

Source NameSales Line

DescriptionRead the Lines of an Order

OperationInsert-Modify

Table Trigger

Permissions2

Allow Requests

Enable

Use-Make Equivalence

FolderSCENARIO 04

Template Use0

Lines

New LineDelete LineUse KeyClear KeyExcel HeaderExcel LineIndent

Target Table ID	Target Table Name	Indentation	Related Table ID	Tag	Operation	Table Trig...	Key	Fields	Formu...	Filters	Fusion	Excel	Ena...	Force (Imp...
→37	Sales Line	0	37	slines	Insert-Mod...			8	No	0	No	0	☒	☐

TableData · 37

Field LinesSearchAnalyzeNewEdit ListDeleteDelete AllGet & Enable AllEnable AllDisable AllCheck DuplicationsEnable SelectedDisable SelectedCheck formulas

Field ID	Field Name	Tag	Order ↑	Series No.	Get From	Field Value	Mapping (Imports)	Ena... Map	Variable Id	Formula	Con...	Ign... For...	Equal Field No.	Fuse	Cha... Sign (nu...	Vali...	Edit...	Visi...	Ena...
→1	Document Type	doctype	0				0	☐			No	☐	0		☐	☐	☐	☐	☒
3	Document No.	docno	0				0	☐			No	☐	0		☐	☐	☐	☐	☒
4	Line No.	line	0				0	☐			No	☐	0		☐	☐	☐	☐	☒
5	Type	type	0				0	☐			No	☐	0		☐	☐	☐	☐	☒
6	No.	itemno	0				0	☐			No	☐	0		☒	☐	☐	☐	☒
15	Quantity	qty	0				0	☐			No	☐	0		☒	☐	☐	☐	☒
22	Unit Price		0				0	☐			No	☐	0		☐	☐	☐	☐	☒
29	Amount		0				0	☐			No	☐	0		☐	☐	☐	☐	☒

If we look at the **Permissions** field in the header, we will see that there are two users with permissions. We examine them by clicking on the number. We see that permissions for **Pos001** exist; however, the **'Allow Delete (from external apps)'** column is unchecked.

NOW Template Permissions					✓ Saved	
Search Analyze ☐ + New Edit List Delete						
	User		User Group	Allow Delete (from external apps)	Enable	
→	TOLO.SASTRE	⋮		☐	☒	
	POS001			☐	☒	

We check it, and we will see that our Post will execute successfully.

HTTP

OE / Scenarios / DeleteLineCustomerOrder

Save

POST

https://api.businesscentral.dynamics.com/v2.0/ /SandboxDev2/ODataV4/OE_DeleteEntity?company=8586dd27-55e9-ed11-884e-6045bdc8c698

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1 {

2 "templateCode": "SCEN04-READORDRLINES",

3 "filterTable": "slines#doctype;Order%slines#docno;101292%slines#line;20000",

4 "senderUser": "Pos001"

5 }

Body

Cookies

Headers (17)

Test Results

Status: 200 OK

Time: 1154 ms

Size: 974 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

1 {

2 "@odata.context": "https://api.businesscentral.dynamics.com/v2.0/ /SandboxDev2/ODataV4/\$metadata#Edm.String",

3 "value": "Delete Entity was successfully!"

4 }

GET SERIES

There is a special function to retrieve and, if needed, increment after reading (reserve it so that no other process or user can use it) a standard Business Central serial code in case our program or app requires it.

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

- Method:** POST
- URL:** `https://api.businesscentral.dynamics.com/v2.0/[redacted]/OpenEntity_Demo/ODataV4/` (The `OE_GetSerieNo` part of the URL is highlighted with a red box).
- Params:** `?company=3967b449-5ae6-ee11-a201-6045bdacc8c3`
- Body:** JSON format, raw view.

```
1 {
2   "serieCode": "V-PED",
3   "increment": "true"
4 }
```

 (A red arrow points to the `"increment": "true"` line.)
- Response:** 200 OK, 925 ms.
 - Body:** JSON format, pretty view.

```
1 {
2   "@odata.context": "https://api.businesscentral.dynamics.com/v2.0/[redacted]/OpenEntity_Demo/ODataV4/$metadata#Edm.String",
3   "value": "101034"
4 }
```

 (The `"value": "101034"` line is circled in red.)

You can view two demonstrations of reading and writing data at

https://youtu.be/STHDS5Qui_E?si=4Gmx5NH2EJT8-FuH

https://youtu.be/8ncaBC_24YQ?si=wSMBLHD-fvaFF1XS