Spider

Manual

Table of Content

Introduction

This document describes the use of Web Services in Spider. The product contains several Web Service API's used to maintain the systems master data, handle sales and get information about inventory availability through external systems. Examples of external systems are Web Shops, CRM Systems or other software providing relevant information that could be used by the program. By using the web services described below the external system will be allowed to create, update and export master data from the program and thereby avoid extra work. Note that all web services regarding Spider have the prefix Spider in the name.

For general information about Web Services, please check out this information https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/webservices/web-services

Web Services to maintain Master Data

Object Type	Service Name	Function	Description
Codeunit	MasterDataAPI	CreateItem	Create new Item
Codeunit	MasterDataAPI	CreateCustomer	Create new Customer
Page	Item	Read	Read Item data
Page	Item	ReadByRecId	
Page	Item	ReadMultiple	Read data from multiple Items
Page	Item	Update	Update Item
Page	Item	UpdateMultiple	Update data from multiple Items
Page	Item	Delete	Delete Item
Page	Item	Delete_ItemCrossReference	Delete Item Cross Reference
Page	Item	Delete_ItemUnitOfMeasure	Delete Item Unit Of Measure
Page	Customer	Read	Read Customer data
Page	Customer	ReadByRecId	
Page	Customer	ReadMultiple	Read data from multiple Customer
Page	Customer	Update	Update Customer
Page	Customer	UpdateMultiple	Update data from multiple Customer
Page	Customer	Delete	Delete Customer

Web Services to handle Sales

Object Type	Service Name	Function	Description
Codeunit	SalesDocumentAPI	CreateOrder	Create new Order
Codeunit	SalesDocumentAPI	CreateOrderAndCustomer	Create Order and Customer in the same **call**
Codeunit	SalesDocumentAPI	CreateReturnOrder	Create new Return Order
Codeunit	SalesDocumentAPI	DeleteOrder	Deletes Order
Codeunit	SalesDocumentAPI	DeleteReturnOrder	Deletes Return order
Codeunit	SalesDocumentAPI	GetOrderConfirmation	Get Order Confirmation in **Base64**
Codeunit	SalesDocumentAPI	GetReturnOrderConfirmation	Get Return Order Confirmation in **Base64**
Codeunit	SalesDocumentAPI	ReleaseOrder	Releases Order in Dynamics 365 Business Central
Codeunit	SalesDocumentAPI	ReleaseReturnOrder	Releases Return Order in Dynamics 365 Business Central
Page	SalesOrder	Read	Reads data from Order in Dynamics 365 Business Central
Page	SalesOrder	ReadMultiple	Reads data from Multiple orders in Dynamics 365 Business Central
Page	SalesReturnOrder	Read	Reads data from Return Orders in Dynamics 365 Business Central
Page	SalesReturnOrder	ReadMultiple	Reads data from Multiple Return Orders in Dynamics 365 Business Central
Page	SalesShipment	Read	Reads data from Sales Shipment in Dynamics 365 Business Central
Page	SalesShipment	ReadMultiple	Reads data from Multiple Sales Shipments in Dynamics 365 Business Central
Page	SalesReturnReceipt	Read	Reads data from Sales Return Receipt in Dynamics 365 Business Central
Page	SalesReturnReceipt	ReadMultiple	Reads data from Multiple Sales Return Receipts in Dynamics 365 Business Central

Web Services to handle Inventory

Object Type	Object Name	Function	Description

Page	ItemInventory	ReadMultiple	Get Inventory Availability for all Items according to given filters.
Page	ItemOnPurchase	ReadMultiple	Get ItemNo. And expected receipt date from open Purchase Order Lines.
Page	ItemOnSales	ReadMultiple	Get ItemNo. And expected delivery date from open Purchase Order Lines.

Web Services to handle External Systems

Object Type	Service Name	Function	Description
Code Unit	ExternalSystemAPI	GetChangesRecords	Get Changes Records

Getting started

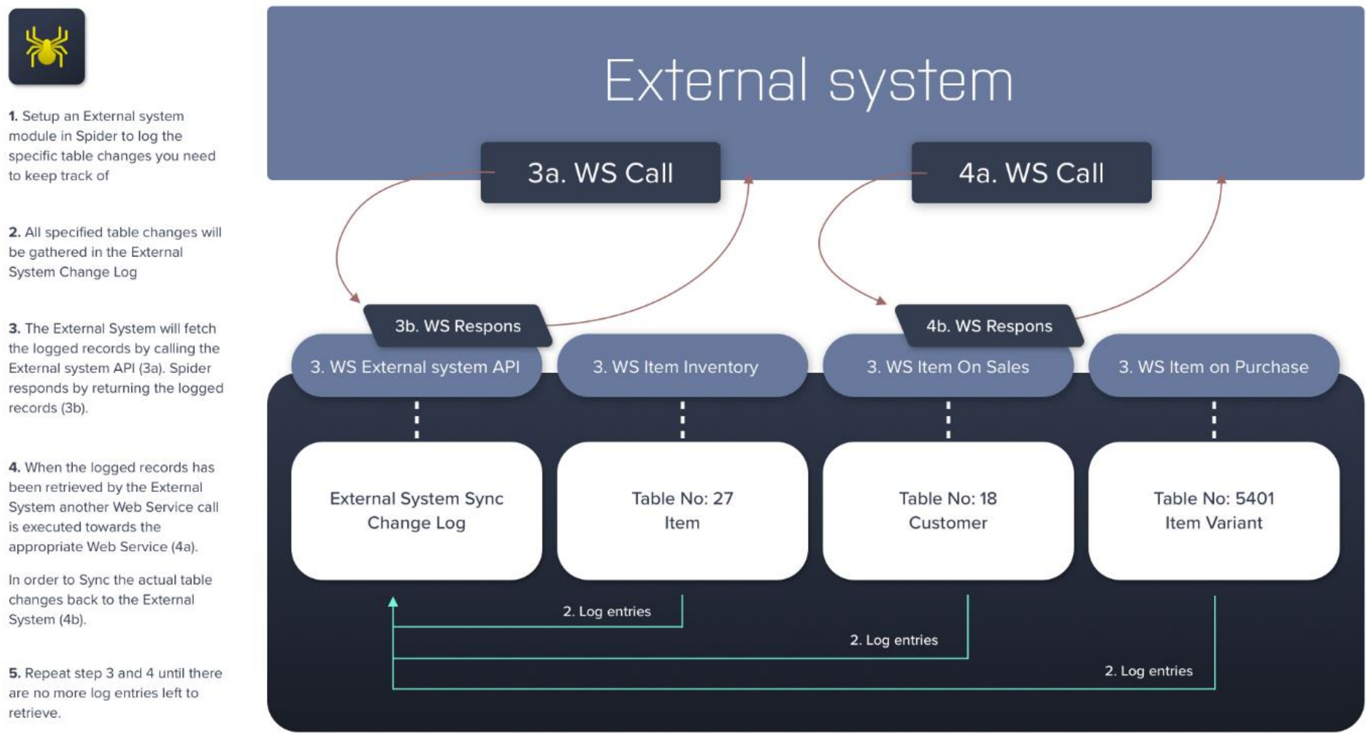
Authentication

For information regarding Web Services Authentication, please visit: https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/webservices/web-services-authentication

External Systems

Overview

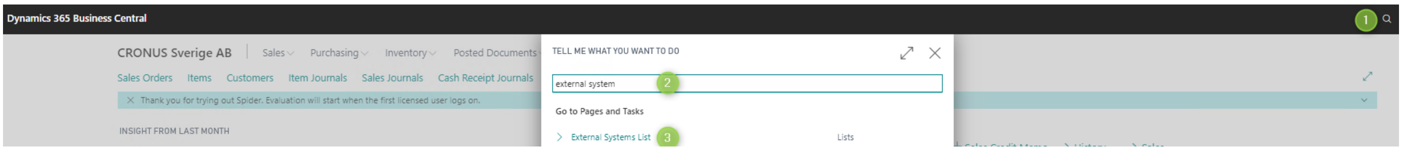
The external systems feature is used when Dynamics 365 Business Central is integrated with other systems. The main purpose of the module is to keep other systems synchronized with Dynamics 365 Business Central by creating log entries. These log entries are then retrieved by the external system via Web Services. Once the log entries are fetched the external system performs a read call on the appropriate Web Service to refresh the data in the external system. The picture below illustrates an example of the intended work flow.



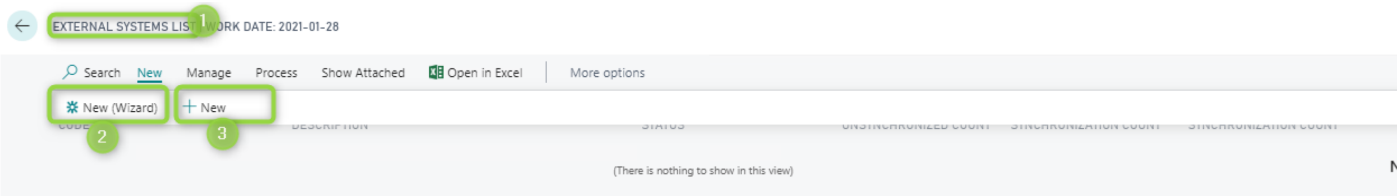
It is possible to log and retrieve logged records from any table in Dynamics 365 Business Central (3). The Web Services described in this document can be used to read data from specific tables and fields (4). If there is a need to synchronize more data than what is offered by the Web Services included in Spider, additional pages can be published from the user interface by running the assisted setup “set up reporting data” in Dynamics 365 Business Central.Setup

Before a Web Service can be used you need to set the communicating counterpart as an External System in your system.

To Navigate in the system, you click on the search icon (1) in the right top corner and enter your search word (2) and click on the link below (3) that matches your wishes.



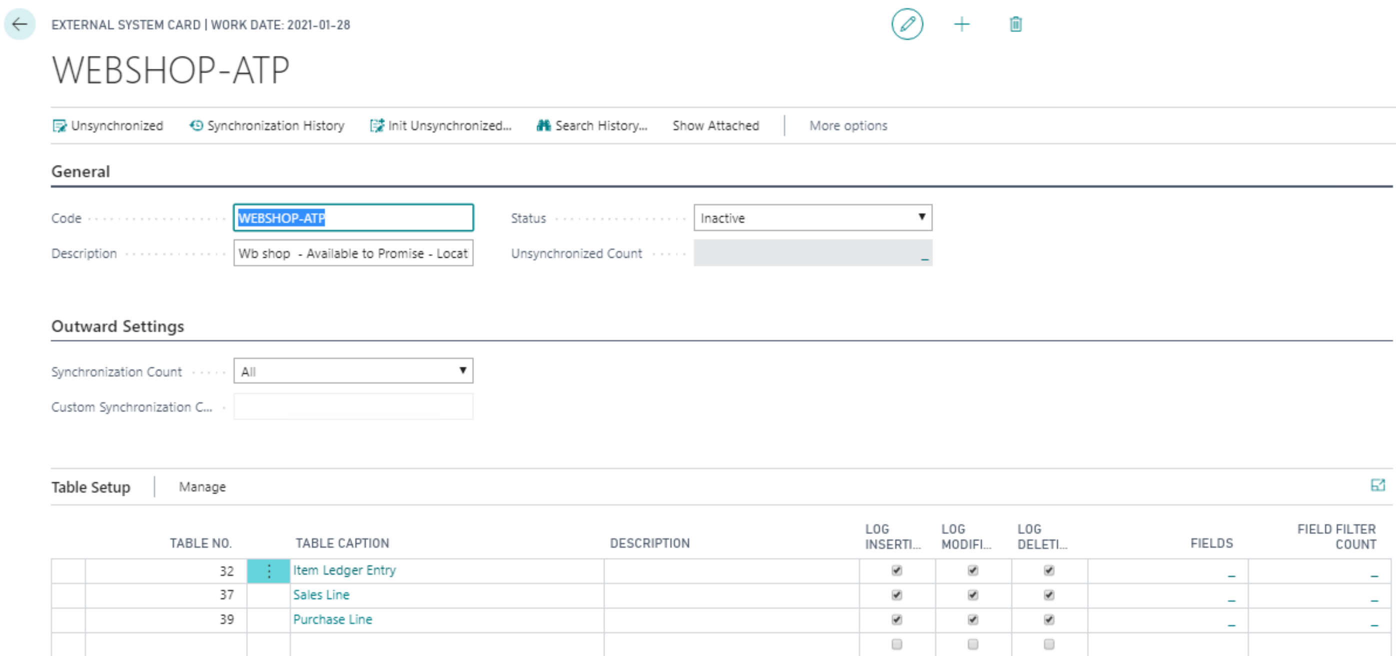
To add an external system, navigate to External Systems list (1) and navigate to new and find “New (Wizard) (2) by adding a new system by using a guide or click New (3) to add an external system manually.



Adding an External system manually

An empty card will be opened. Enter a ***Code(1)*** and a ***Description(2).*** Add tables to synchronize in ***Table Setup(3),*** set ***Synchronization count(4)*** to One at a time and change ***Status(5)*** to Active.

WEBSHOPWEW

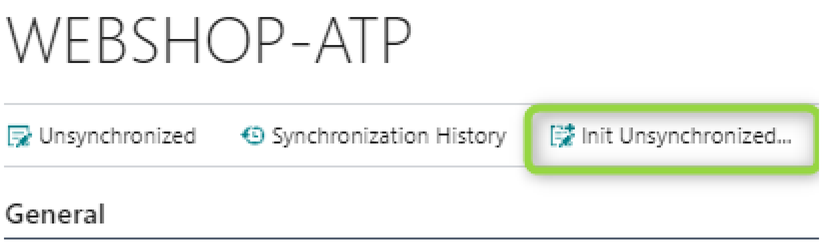


Field	**Function**
Code	Enter the code as a System ID in the API-calls from the external system when communicating with the Dynamics 365 Business Central-database.
Description	Enter a description for the Code.
Status	Available options:

	• Inactive • Active • Under development (used when development is being made to an already active External System.)
Synchronization count	Available options: • One at a time • All
Table Setup	Enter which table to be synchronized, there can be more than one
Table No.	Enter the table that is to be synchronized.
Fields	Add which fields who will trigger log entries.
Field Filter	Apply one or more filters to control which table records that is creating log entries.

Navigation bar functionality

Field	Function
Init Unsynchronized	Creates Log Entries for all records defined in the tables setup. Use this action to perform a Refresh synchronization. The time this will take is depending of the record count of the affected tables.



Field	Function
Unsynchronized	Lists log entries which have not yet been retrieved by the external system.
Synch. History	List over log entries that has been retrieved by the external system.
Synch. History Lines	Detailed list over log entries that has been retrieved by the external system.

Row functionality

To handle specific tables, click on ⋯ on the row.

Field	Function
Fields	Enter fields to trigger logging.
Field Filter	Use to filter records on specific fields to trigger logging.
Init Synchronized	Fills Unsynchronized List with changes regarding chosen table to synchronize.

Web Services

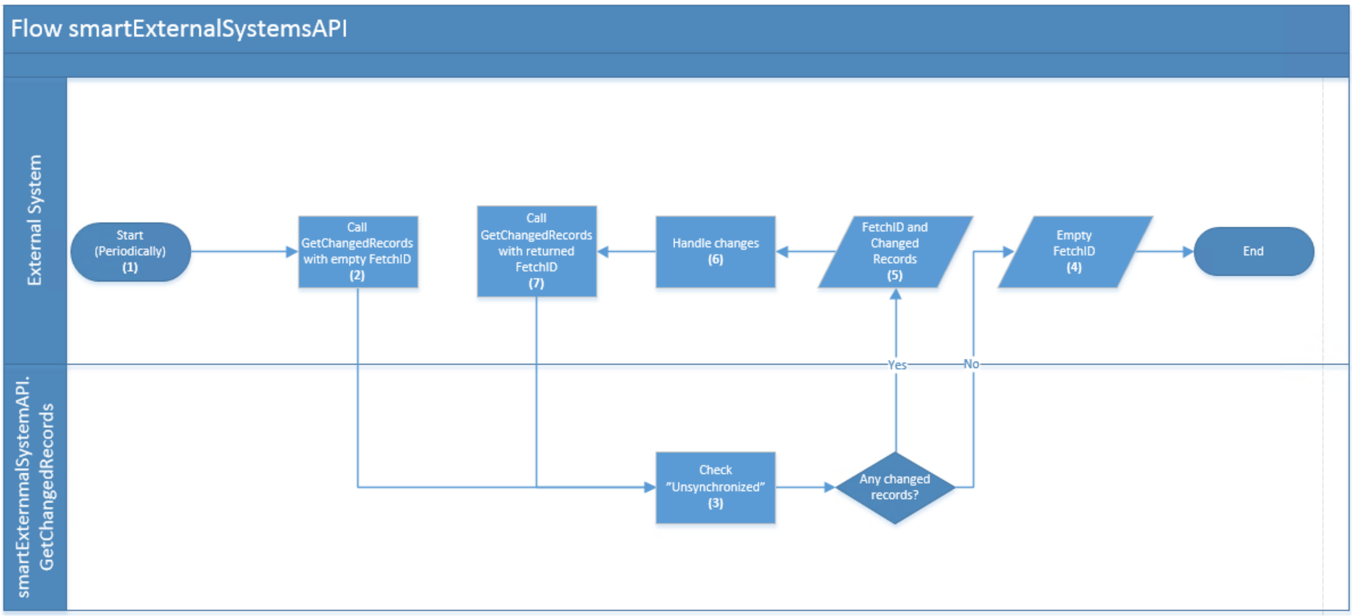
To access SOAP-URL:s being used to access the web services, navigate to the Web Service page (1) The column SOAP-URL shows the address to available API:s. (2) Note: All web services used by Spider have names that starts with Spider.

WEB SERVICES | WORK DATE: 2021-01-28

Spider | + New | Edit List | Delete | Reload | Show Attached | Open in Excel | Actions | Fewer options

OBJECT TYPE	OBJECT ID	OBJECT NAME	SERVICE NAME	ALL TENANTS	PUBLISHED	ODATA V4 URL	ODATA URL	SOAP URL
Codeunit	70314113	QWESR WS External System API	spiderExternalSystemAPI	☐	☑	[illegible]	[illegible]	[illegible]
Codeunit	70314115	QWESR WS Master Data API	spiderMasterDataAPI	☐	☑	[illegible]	[illegible]	[illegible]
Codeunit	70314107	QWESR WS Sales Document API	spiderSalesDocumentAPI	☐	☑	[illegible]	[illegible]	[illegible]
Page	70314120	QWESR WS Customer	spiderCustomer	☐	☑	[illegible]	[illegible]	[illegible]
Page	70314113	QWESR WS Item	spiderItem	☐	☑	[illegible]	[illegible]	[illegible]
Page	70314117	QWESR WS Item Inventory	spiderItemInventory	☐	☑	[illegible]	[illegible]	[illegible]
Page	70314119	QWESR WS Item On Purchase	spiderItemOnPurchase	☐	☑	[illegible]	[illegible]	[illegible]
Page	70314118	QWESR WS Item On Sales	spiderItemOnSales	☐	☑	[illegible]	[illegible]	[illegible]
Page	70314111	QWESR WS S. Return Receipt	spiderPostedSalesReturnReceipt	☐	☑	[illegible]	[illegible]	[illegible]
Page	70314109	QWESR WS Sales Shipment	spiderPostedSalesShipment	☐	☑	[illegible]	[illegible]	[illegible]
Page	70314105	QWESR WS Sales Order	spiderSalesOrder	☐	☑	[illegible]	[illegible]	[illegible]
Page	70314107	QWESR WS Sales Return Order	spiderSalesReturnOrder	☐	☑	[illegible]	[illegible]	[illegible]

Web Services to handle External Systems



1) This flow needs to be repeated frequently, perhaps with the help of a scheduler.
2) External system calls SpiderExternalSystemAPI.GetChangedRecords with an empty Fetch ID and asks if there is any changed records.
3) SpiderExternalSystemAPI.GetChangedRecords checks the Unsynchronized list for changed records.
4) If SpiderExternalSystemAPI.GetChangedRecords does not find changes Fetch ID will be returned empty and the External System knows there are no changed records and the call will end.
5) If SpiderExternalSystemAPI.GetChangedRecords does find changes Fetch ID will be given a GUID to pack the changes and returns the Fetch ID and key of changed records.
6) External system calls relevant Web service to get the changes
7) External system calls SpiderExternalSystemAPI.GetChangedRecords with a Fetch ID and asks if there are any changed records. Step 3 to 7 will repeat until there are no changed records.

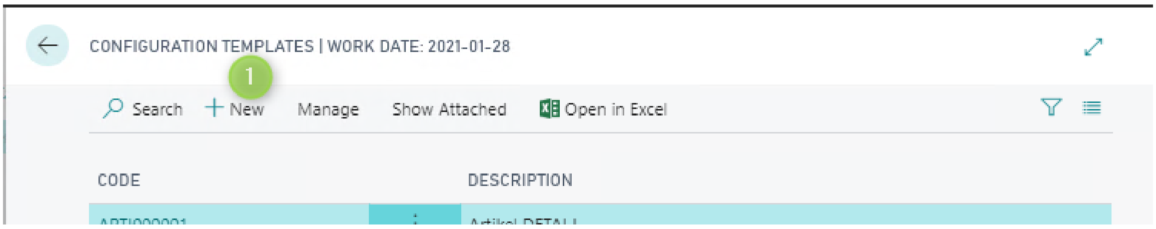
Set Up Customer/Item Templates

When creating a new Customer or Item it is possible to get help with by using templates. Using templates is a way to fill fields that usually are the same within a category. For example, maybe 95% of all domestic customer have the same Payment terms, then it is possible to set up a template for all domestic customer with that Payment Terms Code and other things they have in common. The same thing goes for Items.

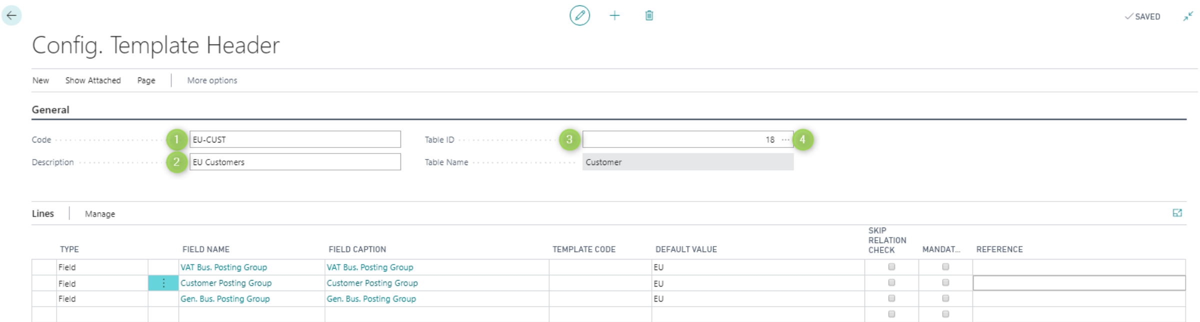
To add a new template:

1. Search (ALT+Q) for Configuration Templates

2. Select New (1)



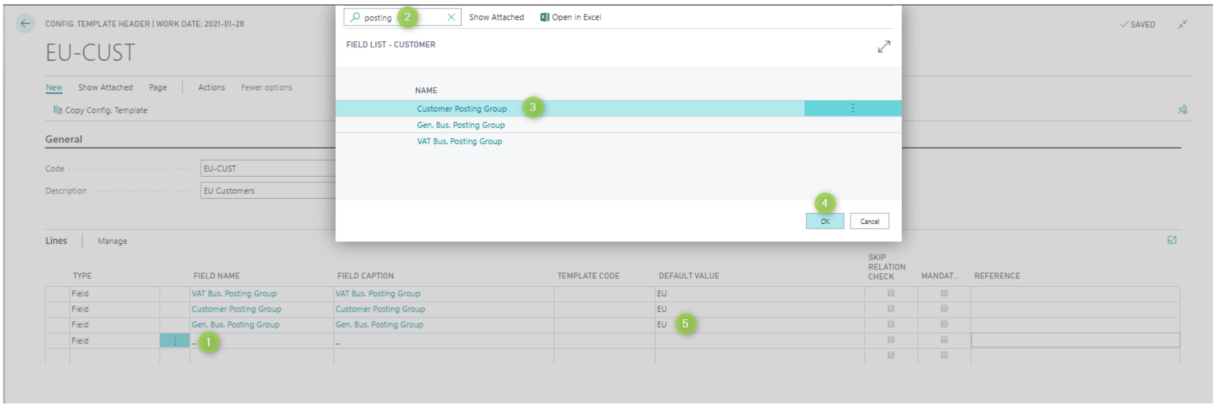
3. An empty card will be opened. Enter a *Code (1)*, a *Description (2)* and *Table ID (3)*.



General Tab

Field	Function
Code	Enter a code as a unique ID for the template.
Description	Enter a description for the Code
Table ID	Enter the table to witch this template applies. The *Table Name* field is automatically filled in when the Table ID field is set. Open the Table list by pressing the button **…**(4)

To set up which fields to fill automatically left click on a new row in the filed name column(1) to open the page with all available fields. Scroll the list or search (2) and select a field (3), exit by selecting OK (4). After add a default value (5) repeat above steps to add more fields.



Extra Fields Setup

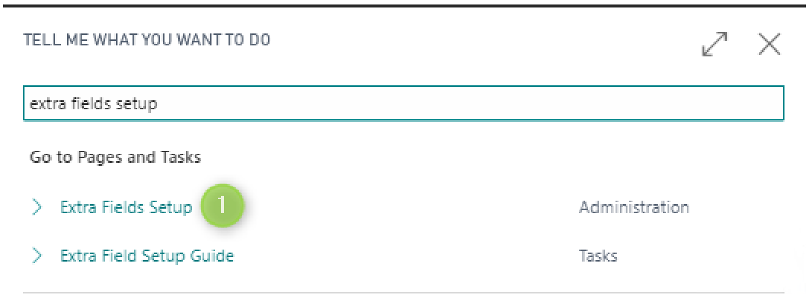
On the Customer card and Item card it is possible to add extra fields. The fields contain no logic but are useful to set filters and store information. Both Customer and Item cards have 28 extra fields spread over 6 different kinds of datatypes, 8 Code, 4 Text, 4 Decimal, 4 Integer, 4 Date and 4 Boolean.

Fields Types

Datatype	Description	Length
Code	This data type stores a special type of string that is converted to uppercase and removes any trailing or leading spaces.	20
Text	This data type stores a text string.	50
Decimal	This data type stores **decimal numbers** ranging from -999,999,999,999,999.99 to +999,999,999,999,999.99.	
Integer	This data type stores **whole numbers** with values that range from -2,147,483,647 to 2,147,483,647.	
Date	Stores a date ranging from January 1, 1753 to December 31, 9999.	
Boolean	Indicates **true** or **false**.	

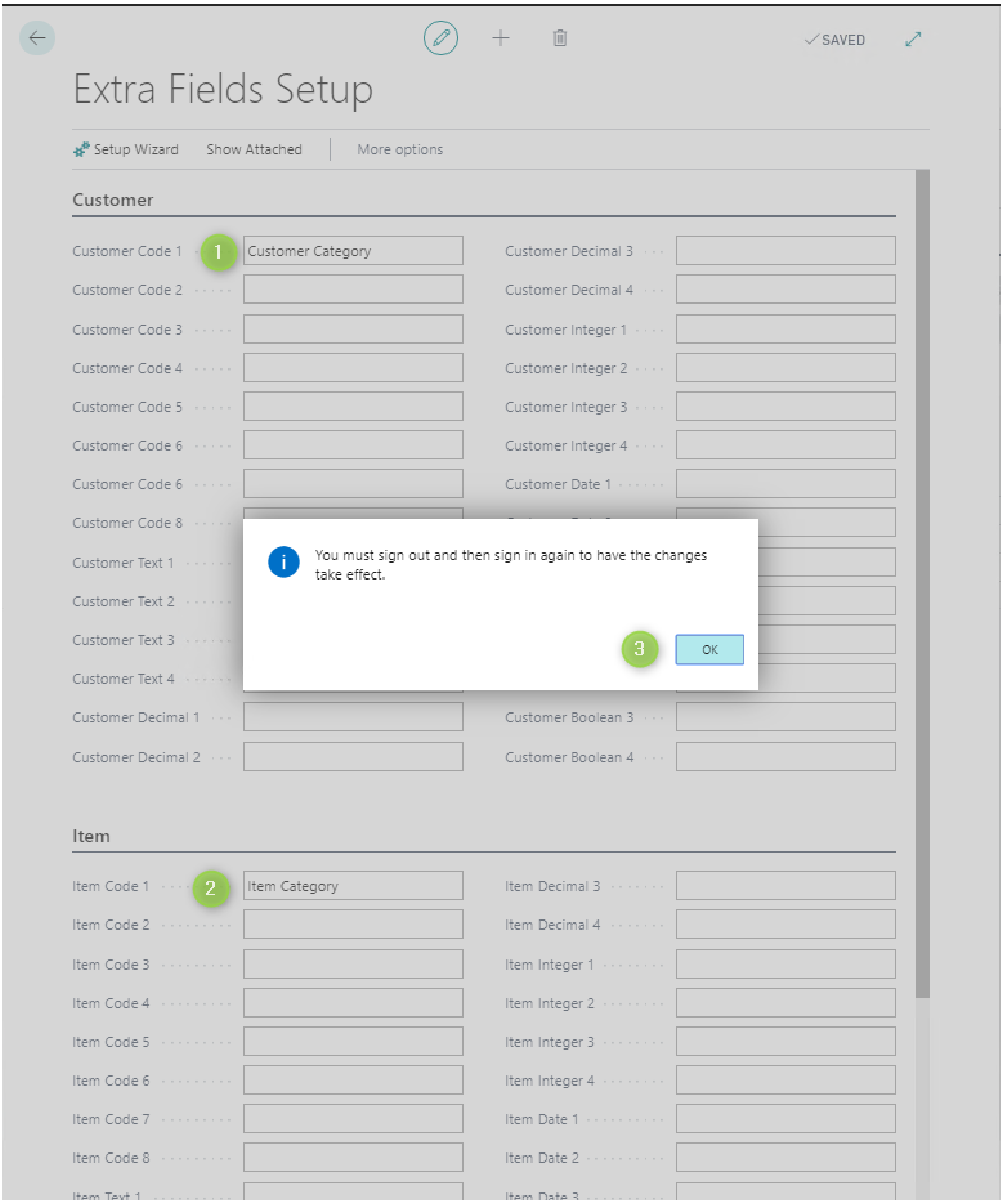
Set up extra fields

1. Search (ALT+Q) for Extra Field Setup and select it (1)

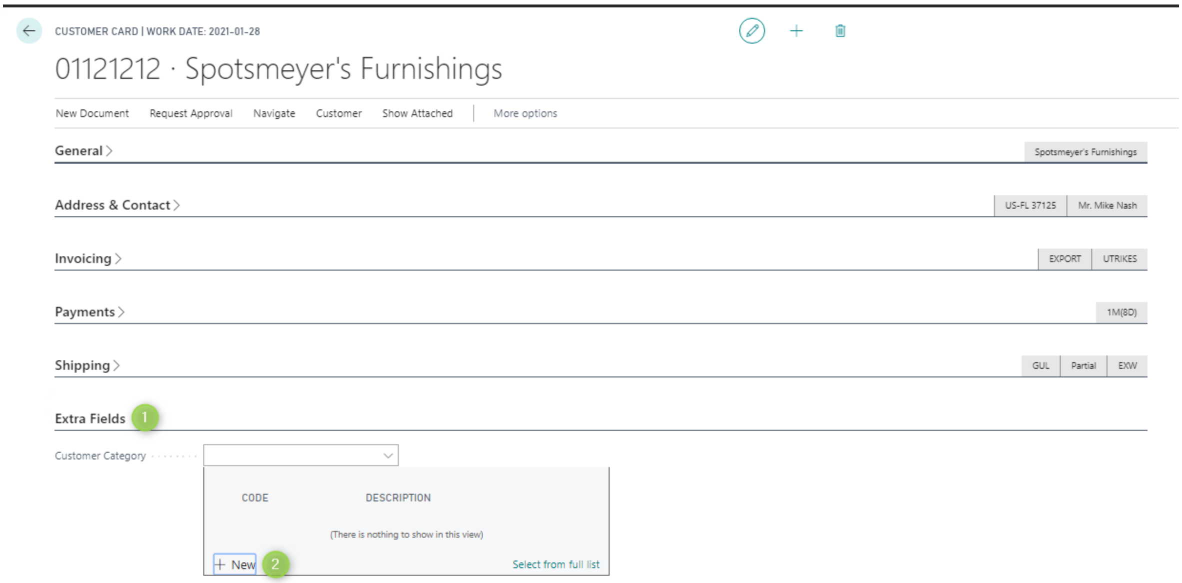


2. Rename the fields to your own standard (1,2) and sign out/in (3) before opening your Customer or Item card

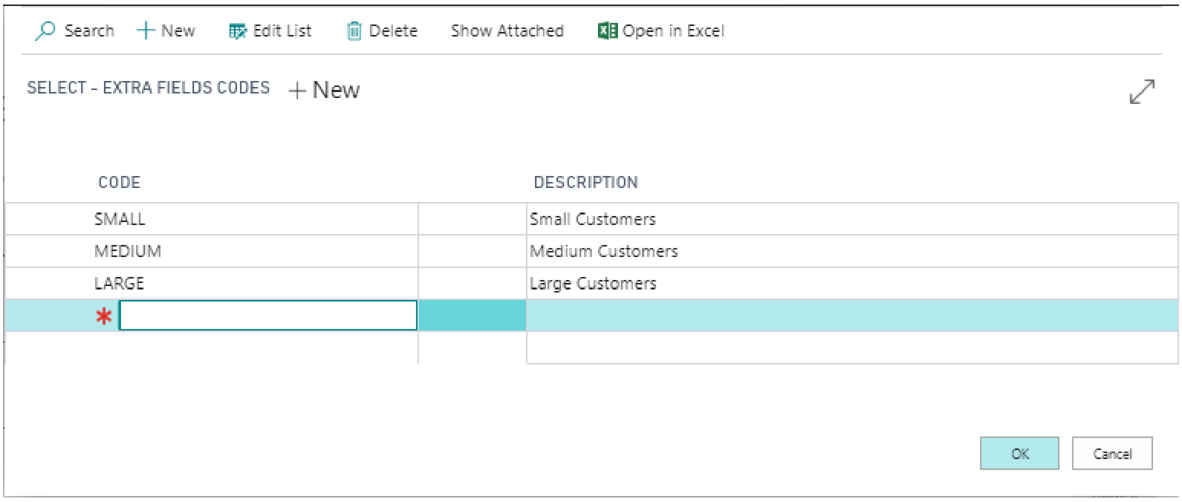
.



3. To sign out and in, reload, press ctrl+F5.
4. You will find all Extra fields under the tab Extra fields (1) at the bottom of your Customer or Item card. Select New (2) to add your own values.



5. Type in your own values and descriptions



LicenseProvider

LicenseProvider manages the billing of all apps from SmartApps 4D365.
All apps from SmartApps can be found and managed in the "SmartApps overview" page.
To get help regarding LicensProvider, please click on the following link:

https://www.smartappsford365.com/wp-content/uploads/LicenseProvider_Manual.pdf