

SpreadsheetDatastoreWriter

Der [SpreadsheetDatastoreWriter](#) kann ein **Spreadsheet**, also tabellarische Daten, in einen Datastore importieren. Dabei kann man wählen ob man nur neue Datensätze einfügen, nur vorhandene aktualisieren oder beides gleichzeitig machen will.

- 1 [Outputs](#)
- 2 [Schema automatisch erweitern](#)
- 3 [Einsatzgebiete](#)
- 4 [Einfacher Import](#)
 - 4.1 [Spaltenbezeichnung bereits im Spreadsheet vorhanden](#)
 - 4.2 [identifizier-Spalten manuell definieren](#)
- 5 [Potentielle Fehlerquellen und Fehlermeldungen](#)
- 6 [Erweiterter Import](#)
 - 6.1 [Identifizier2 Spalte](#)
- 7 [Import von hierarchischen Spreadsheets \(Parent - Variant innerhalb des gleichen Datastores\)](#)

Outputs

Der Step gibt mehrere [Outputs](#) als Spreadsheet zurück:

- neu eingefügten Datensätze,
- die aktualisierten (vorhandenen) Datensätze (egal ob inhaltliche Änderung oder nicht)
- die aktualisierten (vorhandenen) Datensätze mit Inhaltsänderung (bei denen sich mindestens ein Feld inhaltlich geändert hat)
- und die nicht angefassten Datensätze (Datensätze, die im Datastore vorhanden sind, aber nicht (mehr) im zu importierenden Spreadsheet).

Mit diesem Output-Spreadsheets kann man gezielt bestimmte Daten für Folgeprozesse verwenden. Zum Beispiel kann man alle Datensätze, die sich inhaltlich geändert haben (siehe Output "*aktualisierten (vorhandenen) Datensätze mit Inhaltsänderung*") in eine Datei schreiben und exportieren.

Schema automatisch erweitern



Datastore und Schema anlegen / erweitern

Wenn der Datastore vor dem Import noch nicht existiert, dann wird dieser automatisch zusammen mit einem Schema angelegt. Das Schema bekommt dann automatisch alle Spalten des zu importierenden Spreadsheets. Über die Option [autoExtendSchema](#) kann das Schema automatisch erweitert werden, sobald neue Spalten im Spreadsheet dazu kommen.

Einsatzgebiete

Der Step [SpreadsheetDatastoreWriter](#) befindet sich meist am Ende eines Flows bzw. nach einem [SpreadsheetMapper](#), wenn die Daten schlussendlich gespeichert werden sollen. Ein typischer Flow sieht ungefähr so aus:



Einfacher Import

Damit etwas importiert werden kann, muss das **Spreadsheet** mindestens über folgende zwei Spalten verfügen (siehe [Datastores](#)):

- **identifier**
- **folder** (*optional: wenn leer wird automatisch "default" gesetzt*)



Schreibweise beachten

Achten Sie darauf, dass die beiden Spalten exakt so geschrieben werden (alles Kleinbuchstaben).

Über diese zwei Spalten wird jeder Datensatz eindeutig identifiziert. Vorstellen kann man sich das mit Verzeichnissen und Dateien in einem Dateisystem. *Folder* entspricht einem Verzeichnis und *identifier* entspricht der Datei.

Ein *identifier* darf pro *folder* nur 1x verwendet werden. Ansonsten gibt es einen Fehler. Mit anderen Worten: Ein Datensatz ist eindeutig durch *folder* + *identifier*.

Es gibt zwei Varianten, wie man dem Importer diese Spalten mitteilen kann.

Spaltenbezeichnung bereits im Spreadsheet vorhanden

Wenn das zu importierende Spreadsheet min. zwei Spalten mit diesen Spaltenname (*identifier*, *folder*) hat, dann kann das Spreadsheet importiert werden.

Sollte das Spreadsheet diese Spalten noch nicht haben, könnte man mit dem [Mapper SpreadsheetMapper](#) dieser Spalte erzeugen.

folder	identifier	identifier2	name	brandname
	1		Super Shoe XI	K-Star
	2		Vintage	3-Stripes
	3		Infinity	Runnerschoice
	4		Predator v5	Big cat

identifier-Spalten manuell definieren

Alternativ kann man bei der Konfiguration des SpreadsheetDatastoreWriter auch angeben, in welchen Spalten sich *identifier* und *folder* befinden.

identifierColumn



folderColumn



Potentielle Fehlerquellen und Fehlermeldungen



Achtung

Es dürfen dort nur die Spaltennamen des Input-Spreadsheets eingetragen werden, die jeweils den *identifier* bzw. den *folder* beinhalten. Sie können *identifierColumn* und *folderColumn* auch leer lassen, allerdings muss dann das Input-Spreadsheet die beiden Spalten *identifier* und *folder* haben.

Bitte achten Sie darauf, dass Sie **nicht fälschlicherweise** den Name eines Datastore *folders* bei *folderColumn* eintragen.



Sollte das zu importierende Spreadsheet keine identifier-Spalte haben (z.B. weil Sie vergessen haben eine zu definieren), dann bricht der Flow mit Fehlermeldung ab:

Fall 1: No 'Artikelnummer' column in Spreadsheet. This column should be present and filled with a unique identifier of the record (e.g. productID).

- hier wurde `identifierColumn=Artikelnummer` definiert, aber das Spreadsheet hat gar keine Spalte mit dem Namen `Artikelnummer`
- **Lösung:** fügen Sie dem Spreadsheet eine Spalte mit dem Namen 'Artikelnummer' hinzu, oder korrigieren Sie den Wert von `identifierColumn`

Fall 2: No 'identifier' column in Spreadsheet. This column should be present and filled with a unique identifier of the record (e.g. productID).

- hier wurde `identifierColumn` vermutlich leer gelassen, weshalb der Import eine Spalte mit dem Namen 'identifier' erwartet. Diese scheint im Spreadsheet aber nicht vorhanden zu sein.
- **Lösung:**
 - a) fügen Sie dem Input-Spreadsheet eine Spalte mit dem Namen 'identifier' hinzu,
 - b) oder: definieren Sie per `identifierColumn` eine Spalte, die einen eindeutigen identifier enthält (z.B. eine EAN)
 - falls Sie eine Mappingdefinition direkt im SpreadsheetDatastoreWriter Step definiert haben (z.B. einmal auf Configure gedrückt), dann wird die Spalte darin erwartet. Ansonsten muss die Spalte im Input-Spreadsheet vorhanden sein.

Erweiterter Import

Der SpreadsheetDatastoreWriter bietet noch weitere Optionen, um komplexere Import-Szenarien abzubilden.

Identifier2 Spalte

Jeder Datensatz in einem Datastores verfügt über eine `identifier2` Spalte. Diese Spalte kann mit einer weiteren ID gefüllt werden. Diesen `identifier2` kann man später für [Querverweise](#) benutzen.



identifier2 nicht eindeutig

Wichtig: Der `identifier2` muss nicht eindeutig sein. D.h. er kann doppelt vergeben werden. Sie müssen selbst darauf achten, dass dieser eindeutig ist. Ansonsten kann es bei [Querverweisen](#) zu Fehlern kommen.

Es gibt wieder 2 Varianten, wie die `identifier2` Spalte definiert werden kann:

- Spalten `identifier2` bereits im Spreadsheet vorhanden
- oder Sie geben über das Inputfeld `identifier2Column` den Spaltennamen an, der den `identifier2` enthält.

identifierColumn	<input type="text" value="artikelnummer"/>	
folderColumn	<input type="text" value="marke"/>	
identifier2Column	<input type="text" value="lieferantenartikelnummer"/>	



Achtung

Bitte achten Sie auch hier wieder darauf, dass Sie hier Spaltennamen eingeben und nicht fälschlicherweise den Name eines Datastore `folders` bei `folderColumn` eintragen.

Import von hierarchischen Spreadsheets (Parent - Variant innerhalb des gleichen Datastores)

Es besteht die Möglichkeit auch 2-stufige Daten in einen Datastore zu importieren. Das bedeutet, dass es eine Eltern-Zeile (Parent) und dazugehörige Varianten-Zeilen gibt. Man kennt das z.B. bei Produktkatalogen mit Hauptartikel - Artikelvarianten oder bei Rechnungen mit Rechnungskopf - Rechnungspositionen.

Dazu muss im Import-Spreadsheet eine weitere Spalte mit dem Namen **parent_identifier** eingefügt werden. Diese Spalte muss mit dem *identifier* der Parent-Zeile gefüllt werden wie z.B.:

identifier	folder	parent_identifier
1	default	
1a	default	1
1b	default	1

Die Daten werden daraufhin in den Datastore importiert. Die Variant-Datensätze sollten dann an den entsprechenden Parent-Datensätzen hängen.

Wenn der Import korrekt funktioniert hat, dann sollten das in der Datastore-Suche wie folgt aussehen:

Type	Folder	Identifier	Identifier2	name
Master Parent	default	1		Super Shoe XI
(2)				

Daten

[✎ Daten bearbeiten](#)

Feld	Aktueller Wert
name	Super Shoe XI
brandname	K-Star
description	Description of Super Sh
price	129.99
pricecurrency	EUR
stock	0
variants	color=red;size=M

Varianten

Alle Auswählen			name	b
<input type="checkbox"/>	View	Unmatched	Super Shoe XI Variant1	
<input type="checkbox"/>	View	Unmatched	Super Shoe XI Variant2	



Hinweis

Datensätze mit Parent-Variant Beziehung basieren auf der Annahme, dass Parent-Datensatz und Variant-Datensatz [im gleichen Datastore und im gleichen Folder](#) liegen.