

Medien und andere Daten mit Feeds migrieren

Endlich komme ich dazu, meine Webseite »Gartenstauden.de« von Drupal 7 auf Drupal 10 zu migrieren. Die Seite mit 7000 Pflanzen-Portraits begleitet mich seit fast 25 Jahren und hat schon viele Techniken durchlaufen.

Inzwischen ist es eine erfolgreiche Affiliate-Webseite mit Links zu Partner-Baumschulen.

Nicht nur aus technischen Gründen ist ein Relaunch dringend notwendig. Die Seite wird sauber unter Drupal 10 neu aufgesetzt und die vielen Tausend Seiten Inhalte aus ca. 20 Inhaltstypen werden per Feeds importiert.

So gesehen alles nichts Neues und es gibt schon einen Blogbeitrag zum Thema Feeds:

<https://www.montviso.de/blog/nodes-und-taxonomies-von-drupal-7-auf-drup...>

In diesem Fall ist neu, dass ich die Bilder aus der Drupal 7 Version künftig als Medien organisieren und auch diese per Feeds importieren möchte.

Und wie immer gibt es den einen oder anderen Fallstrick, über den ich in Zukunft nicht noch mal stolpern möchte, deshalb dieser Blogbeitrag für mich und alle, die ihn lesen möchten. ;-)

Ich erläutere das Vorgehen am Beispiel Blog, weil der nicht so viele Felder hat, wie die Pflanzen-Inhaltstypen mit allen Kategorien. Die besonders wichtigen Punkte sind mit einem roten Pfeil oder Kreis kenntlich gemacht.

1. Medientyp anlegen, in dem die Bilder künftig gespeichert werden

Es gibt zwei Typen, einmal Vorschaubilder für Facebook und einmal eine Bildergalerie.

Hier der Aufbau der Felder für die Bildergalerie.

Da auch Wikipedia-Bilder verwendet werden, die einen Quellennachweis benötigen, habe ich ein solches Feld eingerichtet.

Startseite > Verwaltung > Struktur > Medientypen > Bearbeite Bilderer-Galerie Blog

Felder verwalten

Bearbeiten

Felder verwalten

Formularanzeige verwalten

Anzeige verwalten

Berechtigungen verwalten

+ Ein neues Feld erstellen

+ Ein vorhandenes Feld wiederverwenden

Beschriftung	Systemname	Feldtyp
Bild	field_media_image_4	Bild
Feeds item	feeds_item	Newsfeed Referenztyp: Newsfeed
Quelle	field_quelle	Text (unformatiert, lang)

2. Inhaltstyp anlegen mit Referenz auf die Medien

Die beiden Referenzen unterscheiden sich in dem Punkt, dass die Anzahl bei Vorschaubild auf eins begrenzt ist, wogegen bei der Bildergalerie unendliche Anzahl eingestellt ist.

Felder verwalten

[Bearbeiten](#)
[Felder verwalten](#)
[Formularanzeige verwalten](#)
[Anzeige verwalten](#)
[Berechtigungen verwalten](#)

[+ Ein neues Feld erstellen](#)
[+ Ein vorhandenes Feld wiederverwenden](#)

Beschriftung	Systemname	Feldtyp
Bilder-Galerie	field_bilder_galerie	Entitätsreferenzierung Referenztyp: Medien Medientyp: Bilderer-Galerie Blog
Body	body	Text (formatiert, lang, mit Zusammenfassung)
Feeds item	feeds_item	Newsfeed Referenztyp: Newsfeed
Thema	field_blog_tags	Entitätsreferenzierung Referenztyp: Taxonomie-Begriff Vokabular: Blog
Vorschaubild	field_vorschau_bild	Entitätsreferenzierung Referenztyp: Medien Medientyp: Vorschaubild

Bilder-Galerie Einstellungen für *Blogeintrag*

[Bearbeiten](#)
[inhalt felder übersetzen](#)

Beschriftung *

Bilder-Galerie

Field Storage

Diese Einstellungen gelten für das Feld *Bilder-Galerie* überall, wo es verwendet wird. Einige wirken sich auch auf die Art und Weise aus, wie die Da sobald die Daten erstellt wurden.

Zu referenzierender Elementtyp *

Medien

Erlaubte Anzahl von Werten

Unbegrenzt

Referenzmethode*
Standard

☒ Referenzierte Entitäten erstellen, sofern diese nicht bereits existieren

Medientyp*

- ☐ Audio
- ☐ Bild
- ☐ Bilderer-Galerie Baumschule
- ☒ Bilderer-Galerie Blog
- ☐ Bilderer-Galerie Stauden
- ☐ Blog Inline
- ☐ Dokument
- ☐ Extern gehostetes Video
- ☐ Headerbild
- ☐ Inline-Dokumente
- ☐ Schmetterlingbilder
- ☐ Video
- ☐ Vorschaubild

3. Export unter Drupal 7

Auf der alten Installation legen wir eine Export View im CSV-Format an. Dafür wird das Modul Views Data Export benötigt. Sobald dieses aktiviert ist, kann man bei der View vom Typ »INhalt« ein neues Display vom Typ »Data Export« anlegen. Als Format wählen wir CSV.

Startseite » Verwaltung » Struktur » Views

Export (Inhalt) ⚙

Modify the display(s) of your view below or add new displays.

Displays

Master Page Data export gartenstauden export **blog** +Hinzufügen

▼ Blog details

Display name: blog

TITEL

Titel: Export

FORMAT

Format: CSV file | Einstellungen

Batched export: Nein

Hier die Details der Einstellungen:

A) Gebe einen Pfad mit der Endung ».csv« ein

B) In den Einstellungen für CSV ist vor allem wichtig, dass HTML Tags beibehalten werden.

Andernfalls werden die formatierten Texte später nicht richtig dargestellt.

C) Das Body Feld kann mit Standardeinstellungen bleiben

D) Für die Zusammenfassung binden wir das Body-Feld noch mal ein und wählen dann:

Configure field: Inhalt: Body

For: This views_data_export (override) ▼

☒ Place a colon after the label

☐ Exclude from display
Enable to load this field as hidden. Often used to group fields, or to use as token in another field.

Formatter
Zusammenfassung oder getrimmt ▼

Trimmed limit *
600 Zeichen
If the summary is not set, the trimmed *body* field will be shorter than this character limit.

► **STYLE SETTINGS**

► **REWRITE RESULTS**

► **NO RESULTS BEHAVIOR**

E) Bei dem Feld mit dem Bild für die Galerie belassen wir alles, wie es ist.

Configure field: Inhalt: Thema

For: This views_data_export (override) ▼

Appears in: node.baumschule, node.blog, node.gartenstauden. Also known as: Inhalt: Tags, Inhalt: Thema.

☒ Create a label
Enable to create a label for this field.

Beschriftung
field_blog_tags

☒ Place a colon after the label

☐ Exclude from display
Enable to load this field as hidden. Often used to group fields, or to use as token in another field.

Formatter
Klartext ▼

► **MULTIPLE FIELD SETTINGS**

► **STYLE SETTINGS**

Achte bei allen Feldern mit einer Relation auf Taxonomie darauf, Klartext, statt Link einzustellen.

4. Vorbereitung der CSV Datei für den Import

Wir rufen die View auf, wodurch die CSV Datei (je nach Browser-Einstellungen) in den Downloads gespeichert wird.

Diese Datei öffnen wir nun zur weiteren Verarbeitung in Excel, bzw. in meinem Fall, in Open Office Calc und erzeugen zwei Kopien. Die Originaldatei sichern wir weg, um später immer mal wieder einen Abgleich machen zu können, ob noch alles passt.

Die erste Kopie verwenden wir, um die Bilder für den Import in Media vorzubereiten. Das erkläre ich am Beispiel Bilder-Galerie, weil es komplexer ist. Was wir benötigen, ist eine CSV-Datei, welche die Original-URL des zu importierenden Bildes enthält und einen eindeutigen Namen, sowie die Felder für ALT-Text und Bild-Titel.

Ich verwende folgende Konvention: Name des Bildes ist Name des Blogbeitrags mit einem vorgestellten Zähler. Name ist hier nicht mit der URL zu verwechseln!

Die Titelzeile plus erste Datenzeile sieht jetzt z.B. so aus:

Titelname	field_vorschau_bild
Blog-Titel	,,

Der erste Suchen-und-Ersetzen-Vorgang erzeugt daraus einen String in diesem Format:

Titelname	field_vorschau_bild
Blog-Titel	Original-1.jpg Original-2.jpg Original- 3.jpg

Mit ein paar weiteren Operationen erzeuge ich aus dieser einen Datenzeile vier Zeilen und füge noch zwei Spalten für ALT-Text und Link-Titel hinzu, in denen jeweils der Titel ohne Zähler steht.

Titelname	Alttext	Linktitel	field_vorschau_bild
1 Blog-Titel	Blog-Titel	Blog-Titel	Original-1.jpg
2 Blog-Titel	Blog-Titel	Blog-Titel	Original-1.jpg
3 Blog-Titel	Blog-Titel	Blog-Titel	Original-1.jpg
4 Blog-Titel	Blog-Titel	Blog-Titel	Original-1.jpg

Diese Datei speichern wir unter »blogvorschaubild.csv« im UTF-8-Format.

Ich verwende TAB als Trennzeichen, weil sich damit einfacher zwischen Calc und Texteditor wechseln lässt. Und weil das Zeichen sicher nicht in meinen Daten vorkommt.

4a) Feed-Typ Bilder-Galerie Blog

Wir erstellen unseren ersten Feed-Typ für den Import aller Vorschaubilder, auf die dann beim Einlesen der eigentlichen Blog-Daten referenziert wird.

Denke daran, dass Import-Periode auf AUS gestellt wird, weil wir möchten den Import nur auf Zuruf ausführen und nicht später automatisch durch Cron.

Protokolleinstellungen habe ich ausgestellt, damit das FTP-Verzeichnis nicht voll läuft. Das würde ich nur einstellen, um bestimmte Fehler zu monitoren.

Startseite > Verwaltung > Struktur > Feed types

Bearbeite *Bilderer-Galerie Blog*

[Bearbeiten](#) [Zuordnung](#) [Tamper](#) [Benutzerdefinierte Quellen](#) [Felder verwalten](#) [Formularanzeige verwalten](#)

Grundeinstellungen

Fetcher: Datei hochladen Parser: CSV Processor: Medien

Medientyp*: Bilderer-Galerie Blog

Einstellungen

Fetcher Einstellungen

Parser Einstellungen

Processor Einstellungen

Import period: Aus

Choose how often a feed should be imported. Die Konfiguration von Cron ist erforderlich.

Im nächsten Reiter stellen wir ein, ob wir neue Datensätze eintragen wollen (selbstverständlich) und ob wir vorhandene Datensätze bei einem weiteren Import überschreiben wollen. Ja, das wollen wir.

Parser Einstellungen

Processor Einstellungen

Protokolleinstellungen
Protokollierung aktiviert

Insert new medieneinträge

☒ Insert new medieneinträge
☐ Do not insert new medieneinträge
New medieneinträge will be determined using mappings that are a "unique target".

Vorhandene medieneinträge aktualisieren

☐ Vorhandene medieneinträge nicht aktualisieren
☒ Replace existing medieneinträge
☐ Vorhandene medieneinträge aktualisieren
Vorhandene medieneinträge werden mit Hilfe von Zuordnungen ermittelt, die einzigartig sind.

Zuvor importierte Elemente

Aufbewahren

Wählen Sie aus, was mit Artikeln geschehen soll, die zuvor importiert wurden, jetzt aber nicht mehr im Feed enthalten sind.

medieneinträge ablaufen lassen

Nie

Wählen Sie aus, nach welcher Zeit medieneinträge gelöscht werden sollen.

☐ Eigentümer: Autor des Feeds
Verwenden Sie den Feed-Autor als Eigentümer der zu erstellenden Entitäten.

Eigentümer

Gast (0)

Wählen Sie den Eigentümer der zu erstellenden Entitäten. Lassen Sie Gast leer.

4b) Feed vom Typ Bilderer-Galerie Blog anlegen und Import durchführen

Wähle hier die Datei aus, die Du oben erstellt hast mit jedem Bild in einer extra Zeile.

Bearbeite Bilderer-Galerie Blog

Ansicht **Bearbeiten** Protokolleinträge Löschen Elemente Devel

Titel*

Bilderer-Galerie Blog

The title of this feed, always treated as non-markup plain text

Datei*

1blogvorschau_0.csv (11.34 KB) Entfernen

Eine lokale Datei ausgewählt.

Begrenzer

TAB

Das Zeichen in das Felder in der CSV-Datei trennt.

☐ Keine Überschriften
Auswählen, wenn die CSV-Datei nicht mit einer Überschriftenzeile beginnt. Wenn ausgewählt, müssen die Quellspalten fürs Mapping nach dem Muster '0', '1', '2' etc. benannt werden.

Import-Optionen

Informationen zum Autor

☒ Aktiv
Deaktivieren Sie das obige Kontrollkästchen, um den periodischen Import für diesen Feed zu deaktivieren.

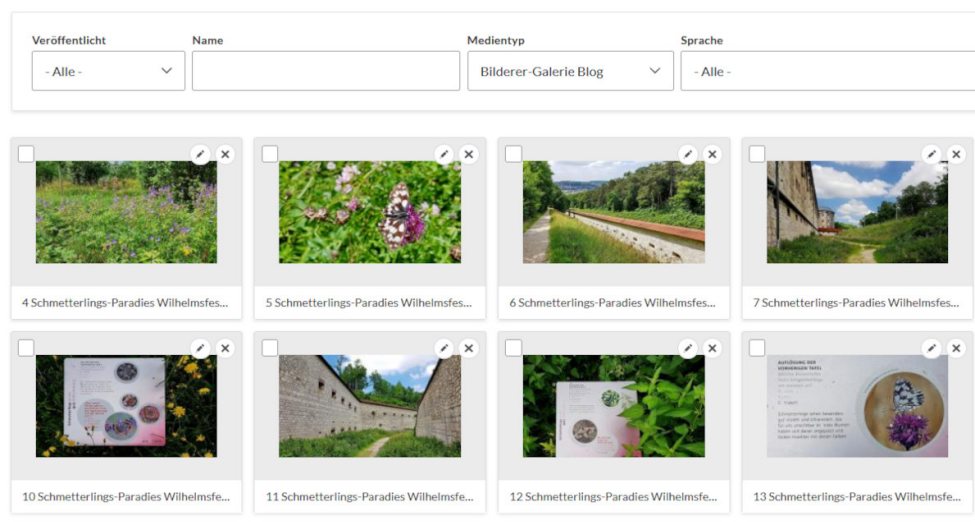
☒ Protokoll
Whether or not logs should be recorded for this feed.

Speichern und importieren **Löschen**

Nun kannst Du den Import laufen lassen. Es gäbe auch die Möglichkeit Import im Background auszuwählen. In dem Fall wird der Vorgang in einem Scheduler gespeichert und erst mit dem nächsten Cron gestartet. Das macht Sinn bei sehr großen Importdateien.

Hier wollen wir das Ergebnis sofort sehen und wählen »Speichern und importieren« aus.

Danach sollten unsere Bilder unter Inhalt -> Medien angezeigt werden:



Das gleiche Prozedere, wie unter 4) machen wir für die Vorschaubilder. Da ist es einfacher, weil nur jeweils ein Bild erlaubt ist.

5. Import der eigentlichen Blog-Daten

Wir haben unter Punkt 3 eine CSV Datei mit den Feldern body, Titel, Zusammenfassung, field_blog_tags, field_vorschau_bild, field_bilder_galerie erzeugt. Ich habe die Feld-Titel mit dem Maschinennamen des Feldes im Inhaltstyp benannt, damit die Zuordnung zwischen Feld in der CSV und Feld in der Datenbanken im Feed leichter fällt. Aber im Grunde ist es egal, wie diese Header heißen.

Der Dateinamen CSV ist in dem Fall nicht ganz korrekt, weil ich kein Komma, sondern den TAB als Trennzeichen verwende, aber auch das hat Gründe. Ich bearbeite die Daten in der Calc-Tabelle nach und halte nebenbei einen

Texteditor mit dieser CSV Datei offen, wo ich geänderte Daten aus Calc einfach rein kopiere. Die haben dann schon den TAB automatisch eingebaut. Andernfalls müsste ich den Speichern-Untern-Vorgang wählen und die Datei jedes mal Im Editor neu öffnen.

Das wäre mühsamer.

Unter Punkt 4 haben wir besprochen, wie wir die Bilder aus Drupal 7 in den gewünschten Media-Typ unter Drupal 10 importieren. Nun müssen wir die Referenz richtig einpflegen. Feed verwendet als Referenz nicht die URL des Bildes, sondern den Namen des Bilder-Mediums.

Da wir jeweils den Titel des Blogs als Namen für das Media-Bild verwenden, kopieren wir diese Spalte in das Feld Vorschaubild. Hier gibt es jeweils nur ein Bild, wir müssen uns also nicht um Mehrfachbezüge und Zähler kümmern.

Dagegen hat das Feld für die Bilder-Galerien mehreren Bilder, und deshalb müssen wir das Prozedere der Umwandlung von der Bild-Source in den Titelnamen hier mit einem Zähler wiederholen.

Aus

```
<img src='Original-1.jpg'>, <img src='Original-2.jpg'>, <img  
src='Original-3.jpg'>
```

wird

```
1 Blog-Titel|2 Blog-Titel|3 Blog-Titel|4 Blog-Titel
```

Nun können wir den Text aus Calc in den Editor kopieren.

Entfernt unten evt. Leerzeichen. Der Cursor muss am Ende der letzten Zeile stehen.

Ein letzter Suchen+Ersetzen-Vorgang ist notwendig, den wir nicht in Calc vornehmen, sondern im Editor:

Wir erinnern uns, dass wir in der Export-View in Drupal 7 für das Zeilenende »xxx« gewünscht haben. Das entfernen wir nun im Editor aus der CSV-Datei.

Nun sollte der endlos lange Quelltext aus dem Body Feld mit HTML-Tags sauber in einer Zeile stehen.

Nennen wir die Datei »blog.csv.«

5a) Feed-Typ Blog erstellen

Wir erstellen den Feed-Typ, wie unter 4a) für den Bild-Import in Media beschrieben.

Nur wählen wir dieses Mal als Processor »Beitrag« und da den Inhaltstyp »Blogeintrag«.

Unter dem Reiter »Zuordnung« findet das Mapping zwischen den Spalten unserer CSV und den Feldern des Inhaltstyps statt.

So erstellen wir eine neue Zuordnung, z.B. für den Titel:
Klicke auf »Select a target«

- Select a target -
Auf der Startseite anzeigen (promote)
Auswirkung auf die Übersetzung der Revision (revision_translation_affected)
Bilder-Galerie (field_bilder_galerie)
Body (body)
Erstellungsdatum der Revision (revision_timestamp)
Feeds item (feeds_item)
Geschrieben am (created)
Geschrieben von (uid)
ID (nid)
Menü-Link (menu_link)
Oben in Listen (sticky)
Protokollnachricht der Revision (revision_log)
Revisions-ID (vid)
Revisionsbenutzer (revision_uid)
Sprache (langcode)
Standardrevision (revision_default)
Standardübersetzung (default_langcode)
Temporary target (temporary_target)
Thema (field_blog_tags)
Titel (title)
URL-Alias (path)
Veröffentlicht (status)
Vorschaubild (field_vorschau_bild)
Übersetzung veraltet (content_translation_outdated)

Zusammenfassung

Konfigurieren

Einde

Format: Einfaches HTML

Referenziert von: Name
Automatisch erstellt Entitäten: Nein

Referenziert von: Name
Automatisch erstellt Entitäten: Nein

Referenziert von: Name
Automatisch erstellt Entitäten: Nein

- Select a target -

Nun entsteht auf der linken Seite ein neues Feld »Select a source«.

- Select a source -

Titel (title)

- Select a target -

Klicke darauf und wähle die erste Option: »Neue Quelle vom Typ CSV column«!
Darunter tragen wir den Feldnamen aus dem Header der CSV-Datei ein.

Neue Quelle vom Typ CSV column

Titel

Geben Sie den genauen CSV-Spaltennamen ein. Dabei wird zwischen Groß- und Kleinschreibung unterschieden.
[titel \[Bearbeiten\]](#)

Titel (title)

Im gleichen Muster erstellen wir die Zuordnungen für die anderen Felder.

Beim body-Feld haben wir hinterher zwei Felder, bei denen wir eine »Neue Quelle vom Typ CSV column« auswählen. Einmal tragen wir »body« ein und einmal den Feldnamen »Zuordnung«.

Am Schluss bitte »Speichern« nicht vergessen!

Wenn Du Dich verklickt hast und ein Mapping entfernen möchtest, klicke einfach die letzte Checkbox an. Dann verschwindet diese Zuordnung sofort.

Wir haben beim Anlegen des Feed-Typs gewünscht, dass nicht nur neue Einträge erzeugt, sondern vorhandene Einträge ersetzt werden sollen. Das ist notwendig, weil auch die größten Optimisten nicht davon ausgehen sollten, dass wir den Import nur einmal machen. In der Praxis tauchen beim Import Fehler auf. Wir verbessern unsere Datenquelle und importieren noch mal. Deshalb muss Feed wissen, welches unserer Felder »unique« ist, also nur einmal vorkommen darf. Dieses Feld ist dann unsere Referenz, die darüber entscheidet, ob wir einen Insert oder einen Update auf einen vorhandenen Datensatz machen möchten.

Im Falle der 7000 Pflanzen, die ich importiert habe, verwende ich jeweils eine Kennzahl, die ich aus der alten Installation übernehme.

Hier wählen wir den Titel durch Auswahl der Checkbox unter »Eindeutig«.

Quelle	Ziel	Zusammenfassung	Konfigurieren	Eindeutig	Eindeutig
body	Body (body): Text	Format: Einfaches HTML	⚙	☐	☐
Zusammenfassung (zusammenfassung)	Body (body): Zusammenfassung				
Titel (title)	Titel (title)		⚙	☒	☒
field_blog_tags	Thema (field_blog_tags)	Referenziert von: Name Automatisch erstellt Entitäten: Nein	⚙		
field_bilder_galerie	Bilder-Galerie (field_bilder_galerie)	Referenziert von: Name Automatisch erstellt Entitäten: Nein	⚙		
field_vorschau_bild (field_vorschau_bild1)	Vorschaubild (field_vorschau_bild)	Referenziert von: Name Automatisch erstellt Entitäten: Nein	⚙		
- Select a target -					

Nun braucht es noch eine wichtige Einstellung unter Body, damit wir nicht später im Frontend den Quelltext, sondern den formatierten Text sehen. Beim Export in Drupal 7 haben wir ausgewählt, dass HTML-Tags erhalten bleiben sollen. Jetzt klicken wir unter »Konfigurieren« auf das Zahnrad und wählen statt »Klartext« das

Text-Format (Filter) aus, das wir für dieses Feld vorgesehen haben.

Quelle	Ziel	Zusammenfassung	Konfigurieren
body	Body (body): Text	Sprache Standard Filterformat Einfaches HTML	Aktualisieren Abbrechen
Zusammenfassung (zusammenfassung)	Body (body): Zusammenfassung		

5c)

Tamper ist ein Modul, das zusätzliche Manipulationen an den Daten während dem Import erlaubt.

Früher musste man ein Extra Modul für die Plugins von Tamper installieren. Heute wird das mit Feeds mitgeliefert, muss aber bei den Modulen aktiviert werden.

Für jede Manipulation steht ein Plugin bereit. In unserem Fall brauchen wir zwei Arten von Manipulationen.

Fall 1: Wir verwenden Referenzen, wie z.B. Taxonomien oder Medien.

Tampers hat einen Bug, der auftritt, wenn in der CSV-Datei ein Feld für eine Taxonomie leer ist.

Da es sich nicht immer um ein Pflichtfeld handelt, kann das vorkommen.

Dieser Bug ist bekannt und wird in den Issues besprochen. Eine Fehlermeldung verhindert dann den weiteren Import dieses Datensatzes.

Wir vermeiden das, indem wir unter Tampers wünschen, dass das nächste Plugin für den Explode für dieses Feld übersprungen wird, wenn das Feld leer ist. Dieses Plugin muss also zuerst durchgeführt werden.

Fall 2: Das zweite Plugin regelt den Umgang mit Mehrfachwerten. Wir haben z.B. bei der Bilder-Galerie mehrere Bildnamen in einem Feld mit PIPE-Zeichen getrennt. Und jetzt sagen wir dem Feed-Import über Tamper, dass dieses Feld am |-Zeichen gesplittet werden soll und entsprechend werden mehrere Bezüge erstellt.

Wir rufen also bei der Pflege des Feed-Typs den Reiter »Tamper« auf und suchen das Feld »field_bilder_galerie« und klicken auf »Plugin hinzufügen« und treffen zuerst diese Auswahl:

Startseite > Verwaltung > Struktur > Feed types > Bearbeite Blog > Tamper

Add a tamper plugin to Blog : field_blog_tags

The plugin to add *

Skip tamper on empty

- Auswählen -

Date/time

String to Unix Timestamp

Unix timestamp to Date

Filter

Erforderlich

Keyword filter

Skip tamper on empty

List

Als zweites Feld wählen wir »Explode«. Die einzige Änderung findet unter »String separator« statt, wo wir das PIPE-Zeichen eintragen, oder was immer wir als Separator gewählt haben.

Startseite > Verwaltung > Struktur > Feed types > Bearbeite Blog > Tamper

Add a tamper plugin to Blog : field_blog_tags

The plugin to add *

Explode

Beschriftung *

Explode

A useful description of what this plugin is doing.

Break up sequenced data into an array

String separator

||

Begrenzung

If limit is set and positive, the returned items will contain a maximum of limit with the last item containing the rest of string. If limit is negative, all components except the last -limit are returned. If the limit parameter this is treated as 1. If limit is not set, then there will be no limit on the number of items returned.

Das sieht dann so aus:

Beschriftung	Beschreibung	Plugin	Aktionen
⛶ Skip tamperers on empty	If it is empty, further Tamper plugins won't be applied.	Skip tamperers on empty	Bearbeiten
⛶ Explode	Break up sequenced data into an array	Explode	Bearbeiten

[Plugin hinzufügen](#)

5c) Feed vom Typ Blog anlegen und Import durchführen

Wir legen unter Inhalt einen neuen Feed vom Typ »Blog« an, wie unter 4b) beschrieben und wählen die unter Punkt 5 erzeugte Datei »blog.csv« aus, laden sie hoch und wählen »Speichern und importieren« .

Vermutlich bekommen wir mehrere Fehler. z.B. könnte ein Feld leer sein, welches Pflichtfeld ist.

In diesem Fall erhalten wir eine qualifizierte Meldung und können entweder den fehlenden Feldwert in unsere CSV eintragen, oder die Angabe »Pflichtfeld« deaktivieren.

Ich hatte das Problem, dass ich beim Import der Pflanzen mit zahlreichen Taxonomien viele Meldungen bekam, weil ich einige Änderungen durchgeführt hatte und vorher Strings im Feld standen, statt Referenzen auf Taxonomien.

Das sah dann so aus:

```
Referenzierten Eintrag für das Feld name mit dem Wert Buch-Rezension
gefunden.
Referenzierten Eintrag für das Feld name mit dem Wert Gewächshaus
gefunden.
Referenzierten Eintrag für das Feld name mit dem Wert xyz gefunden.
...U.S.W...
```

Wunderbar, der Eintrag wurde gefunden, so soll es sein, dachte ich mir.

Hier liegt aber ein Übersetzungsfehler vor. Es sollte heißen:

```
Referenzierten Eintrag für das Feld name mit dem Wert Buch-Rezension
NICHT gefunden.
```

Es hat mich einige Zeit gekostet, bis ich verstand, warum so viele Datensätze nicht eingelesen wurden. Erst, als ich alle Meldungen abgearbeitet hatte, wurde alles korrekt übernommen.

Den Übersetzungsfehler habe ich in den Issues gemeldet und er wurde bereits für künftige Versionen von Feeds geändert.

Bitte beachtet auch Folgendes: Wenn der Import-Vorgang im gleichen Feed wiederholt wird, weil man z.B. einen fehlenden Taxonomie-Eintrag ergänzt hat, oder wenn an der CSV-Datei eine Änderung gemacht wurde, sollte sie unter neuem Namen hoch geladen werden, weil sonst Änderungen nicht erkannt werden. Also statt »blog.csv« verwenden wir dann beim 3. Versuch »3blog.csv«, auch wenn die Datei vielleicht gar nicht geändert wurde, sondern eine Anpassungen im Drupal Backend an den Taxonomien vorgenommen wurde.

Wenn keine weiteren Fehler mehr auftreten und gründlich getestet wurde, gilt unser Blog-Import als geschafft.

Das Vorgehen hört sich schrecklich aufwändig an und ist es auch. Aber verglichen mit dem händischen Import von so vielen Datensätzen ist die Methode elegant und erlaubt auch die Veränderung der Daten in der Calc-Tabelle. So ein Relaunch ist ja selten mit einer 1:1 Übernahme verbunden.

Andere verwenden dafür das Modul Migrate, aber ich finde Feeds angenehmer.