
Inhaltsverzeichnis

Benutzerhandbuch

Einleitung	1.1
Installation und Update	1.2
Grundaufbau	1.3
API-Konfiguration	1.3.1
API-Jobs	1.3.2
Ausführung	1.3.3
Die Standard-API	1.4
Tutorials	1.5
Praxis-Beispiel: Export	1.5.1
API-Konfiguration	1.5.1.1
Job-Konfiguration	1.5.1.2
Ausführung	1.5.1.3
Praxis-Beispiel: Import	1.5.2
API-Konfiguration	1.5.2.1
Job-Konfiguration	1.5.2.2
Ausführen	1.5.2.3
API-Starter Frontend-Modul	1.6
Developer	1.7
Hooks	1.7.1
dataOutput	1.7.1.1
apiComplete	1.7.1.2
"MyAPI" tutorial	1.7.2
File structure	1.7.2.1
Register a new API	1.7.2.2
API configuration	1.7.2.3
"MyCustomCatalogApi" class	1.7.2.4
Test your work	1.7.2.5

CustomCatalog API Interface

Benutzerhandbuch. Stand 16.09.2016

Mit Version 2.0 des [Premium Contao Themes CustomCatalog](#) Moduls (im weiteren Verlauf kurz "CC") wurde eine modulare API Schnittstelle (Interface) hinzugefügt, die es ermöglicht Daten aus dem System (Export) oder in das System (Import) zu führen.

Darüber hinaus bietet der modulare Aufbau Entwicklern die Möglichkeit sowohl die *Standard-API*-Schnittstelle durch eigene Erweiterungen mit neuen Funktionen zu ergänzen, als auch die Integration vollständig autarker "neuer" API-Schnittstellen für das System bereitzustellen.

Dieses Benutzerhandbuch wird im Folgenden zwei praxisnahe Beispiele aufgreifen und damit den Arbeitsprozess "Workflow" im Contao-Backend darstellen.

Praxisbeispiel 1 wird den Export bestehender CC-Datensätze in das allgemeingültige CSV-Format darstellen.

Praxisbeispiel 2 zeigt den Import aus einer Datenquelle (hier die vorab erstellte CSV-Datei).

Ausblick

Dieses Benutzerhandbuch wird im weiteren Verlauf mit einem "Entwickler"-Teil ergänzt. Es wird anhand von Beispiel-Erweiterungen die *Standard-API*-Schnittstelle um neue Logiken ergänzt, als auch eine "neue" API-Schnittstelle in das System integriert.

Hinweis, Disclaimer

Die Nutzung der API-Technologie erfolgt auf eigene Gefahr. Falsch konfigurierte Schnittstellen, Hook-Anbindungen oder Datenquellen jedweder Art können zu Datenverlust führen.

Installation und Updates

Eine Sicherung der aktuellen Daten (Datenbank, Dateien, ...) sollte vor jedem systemeingreifenden Prozess erstellt werden.

Installation

Standalone: [CustomCatalog Installation](#)

[PCT Eclipse Theme](#) + CustomCatalog Bundle Installation: [hier](#).

Update

Vorbereitung

- Laden Sie die aktuelle Version des Moduls über Ihr Benutzerkonto. Updates des CustomElements Moduls finden sich in den Theme-Zips.
- Contaos internen Cache über die Systemwartung leeren oder in den Systemeinstellungen deaktivieren

Schritt 1: Modul ersetzen

Ersetzen Sie den Modul-Ordner "pct_customelements_plugin_customcatalog" im `/system/modules/` Ordner der Contao-Installation (ein Vorheriges Löschen des aktuellen Modul-Ordners garantiert keine doppelten Dateien oder andere Fehler, die seitens des Uploads entstehen können)

Schritt 2: Datenbank-Update durchführen

Öffnen Sie das Contao Backend oder das Contao Installations-Tool und führen Sie ein Datenbank-Update durch. CC 2.0 bringt zwei neue Tabellen in die Contao-Datenbank:

- `tl_pct_customcatalog_api`
- `tl_pct_customcatalog_job`

Und ein neues Feld in der tl_modules Tabelle.

Erweiterungsverwaltung

Zurück

Datenbank aktualisieren

Neue Tabellen anlegen

☐ Alle auswählen
 ☒

```

CREATE TABLE `tl_pct_customcatalog_api` (
  `id` int(10) unsigned NOT NULL auto increment,
  `pid` int(10) unsigned NOT NULL default '0',
  `sorting` int(10) unsigned NOT NULL default '0',
  `tstamp` int(10) unsigned NOT NULL default '0',
  `type` varchar(255) NOT NULL default '',
  `mode` varchar(255) NOT NULL default '',
  `title` varchar(255) NOT NULL default '',
  `description` varchar(255) NOT NULL default '',
  `source` varchar(32) NOT NULL default '',
  `target` varchar(255) NOT NULL default '',
  `tableSRC` varchar(255) NOT NULL default '',
  `singleSRC` binary(16) NULL,
  `templateSRC` varchar(128) NOT NULL default '',
  `hookSRC` varchar(255) NOT NULL default '',
  `filenameLogic` varchar(255) NOT NULL default '',
  `sendToBrowser` char(1) NOT NULL default '',
  `path` binary(16) NULL,
  `uniqueSource` varchar(255) NOT NULL default '',
  `uniqueTarget` varchar(255) NOT NULL default '',
  `allowUpdate` char(1) NOT NULL default '',
  `allowInsert` char(1) NOT NULL default '',
  `allowDelete` char(1) NOT NULL default '',
  `purgeTable` char(1) NOT NULL default '',
  `delimiter` varchar(16) NOT NULL default '',
  `cronjob` char(1) NOT NULL default '',
  `cron` varchar(32) NOT NULL default '',
  `backup` char(1) NOT NULL default '',
  `maxRange` int(5) NOT NULL default '1',
  `isPublished` char(1) NOT NULL default '',
  `published` char(1) NOT NULL default '',
  PRIMARY KEY (`id`),
  KEY `pid` (`pid`)
) ENGINE=MyISAM DEFAULT CHARSET=UTF8;

```

☒

```

CREATE TABLE `tl_pct_customcatalog_job` (
  `id` int(10) unsigned NOT NULL auto increment,
  `pid` int(10) unsigned NOT NULL default '0',
  `sorting` int(10) unsigned NOT NULL default '0',
  `tstamp` int(10) unsigned NOT NULL default '0',
  `type` varchar(128) NOT NULL default '',
  `target` varchar(96) NOT NULL default '',
  `source` varchar(96) NOT NULL default '',
  `mode` varchar(128) NOT NULL default '',
  `reference` varchar(96) NOT NULL default '',
  `code` varchar(255) NOT NULL default '',
  `addTitle` char(1) NOT NULL default '',
  `title` varchar(255) NOT NULL default '',
  `description` varchar(255) NOT NULL default '',
  `attr_id` char(10) NOT NULL default '',
  `valueSRC` varchar(255) NOT NULL default '',
  `hookSRC` varchar(255) NOT NULL default '',
  `sqlSRC` mediumtext NULL,
  `tstampSRC` varchar(10) NOT NULL default '',
  `action` varchar(64) NOT NULL default '',
  `published` char(1) NOT NULL default '',
  `onError` varchar(32) NOT NULL default '',
  `rule` varchar(32) NOT NULL default '',
  PRIMARY KEY (`id`),
  KEY `pid` (`pid`)
) ENGINE=MyISAM DEFAULT CHARSET=UTF8;

```

Neue Spalten anlegen

☐ Alle auswählen
 ☒

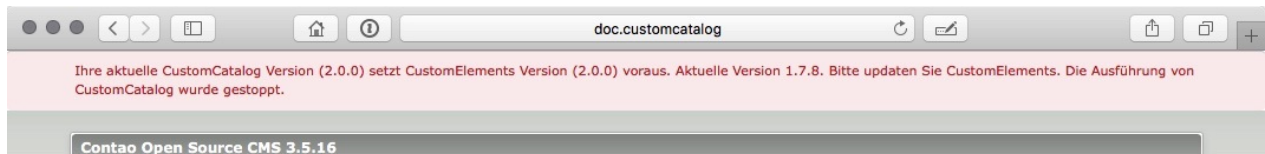
```

ALTER TABLE `tl_module` ADD `customcatalog_api` int(10) NOT NULL default '0';

```

Die Tabellenstruktur kann sich im Laufe der zukünftigen Versionen ändern.

CustomCatalog 2.0 setzt das CustomElements Modul (CE) in Version (min.) 2.0 voraus. CC wird die Verarbeitung stoppen und einen Hinweistext im Contao Backend herausgeben, falls eine frühere CE Version im System gefunden wird.



Changelog

Den changelog jedes Updates findet sich im Premium Contao Themes [Forum](#) im Bereich [Allgemeines](#).

Grundaufbau

Eine CustomCatalog API besteht aus zwei Komponenten:

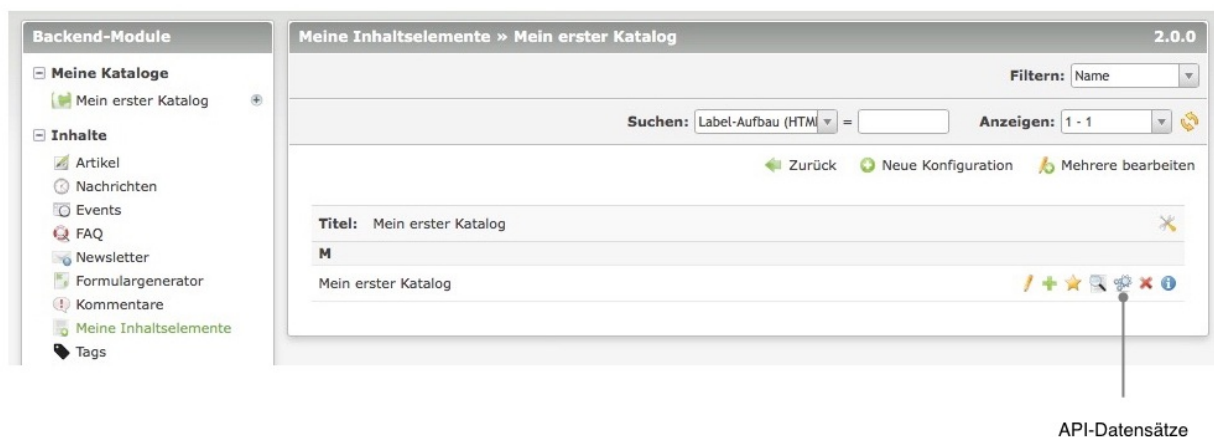
1. [API-Konfiguration](#)
2. [API-Aufgaben \(Jobs\)](#)

Wo finde ich die API-Schnittstellen?

API-Schnittstellen stehen in direkter Beziehung zu den CC-Konfiguration bzw. zu den daraus resultierenden Tabellen. Die Übersicht der aktuellen API-Datensätze erreicht man über das API-Schnittstellen Symbol bei den Operations-Buttons des CC-Konfigurations-Datensatzes.

Dieses Beispiel bezieht sich auf das Tutorial "Mein erster Katalog". [Hier...](#)

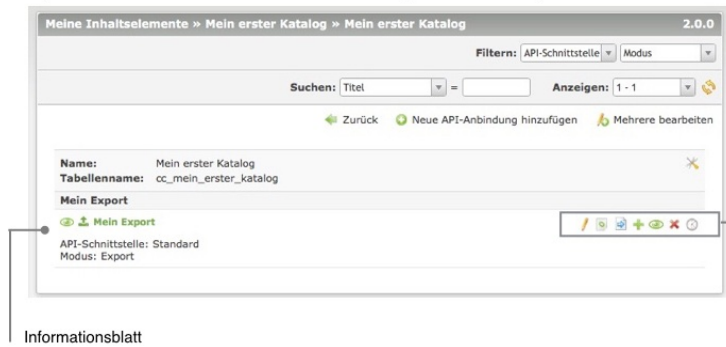
Operations-Buttons der CC-Konfigurationen von CustomElement „Mein erster Katalog“



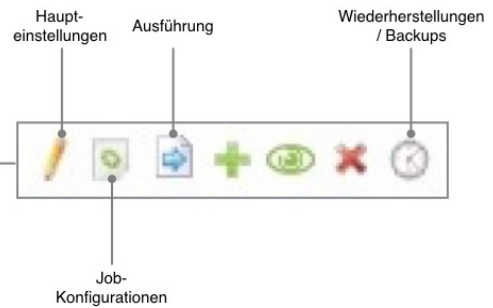
Legende: API-Schnittstellen-Datensätze

Ein API-Datensatz bringt folgende Operationen zusätzlich zu den allgemein bekannten Contao-Operations-Buttons:

Operations-Buttons eines API-Datensatzes vom Typ: Standard > Export



API-bezogene-Operations-Buttons, Detail



- **Haupteinstellung:** Bearbeitet die API-Konfiguration
- **Job-Konfiguration:** Öffnet die Jobs-Übersicht und erlaubt das Anlegen von API-Aufgaben
- **Ausführung:** Startet den API-Prozess im Backend
- **Wiederherstellung/Backups:** Leitet auf Contaos Wiederherstellung (System > Wiederherstellen) und filtert nach API- bzw. CC-Tabellen-bezogenen Wiederherstellungs-Punkten.
- Auge: API aktiviert/deaktiviert
- Grünes Plus: API-Datensatz duplizieren
- Weißes Plus auf grün: Neue API-Anbindung hinzufügen
- Blauer Pfeil, Grüne Balken: Datensatz verschieben (kein Effekt auf API-Ausführungen)

1. API-Konfiguration

Die API-Konfiguration definiert die Art der Schnittstelle. Im Allgemeinen stellt eine API-Schnittstelle einen Betriebsmodus bereit z.B. Export oder Import (Standard-API). Abhängig vom Betriebsmodus stehen individuelle Einstellungsoptionen bereit.

Beispiel: API Konfiguration > Standard-API > Modus: Export

The screenshot displays the 'API-Anbindung 1 bearbeiten' (Edit API Binding 1) configuration page. The interface is divided into several sections with expandable/collapsible headers:

- API-Auswahl** (API Selection): Contains the 'API-Schnittstelle' (API Interface) dropdown set to 'Standard' and the 'Modus' (Mode) dropdown set to 'Export'. A slider control is also visible next to the mode dropdown.
- API-Aufgaben-Einstellungen** (API Task Settings): Includes a 'Titel*' (Title) field with 'Mein Export' and a 'Beschreibung' (Description) field.
- Daten-Einstellungen** (Data Settings): Includes a 'Ziel' (Target) dropdown set to 'CSV-Export', a 'Trennzeichen' (Delimiter) dropdown set to 'Semikolon [;]', and a 'Dateibezeichnung' (File Name) field with the value 'meinExport_{table}'. There is also a 'Zielpfad' (Target Path) section with an 'Auswahl ändern' (Change selection) button and a checkbox for 'An Browser senden' (Send to browser).
- Experten-Einstellungen** (Expert Settings): Includes a checked 'Aktivieren' (Activate) checkbox.

At the bottom, there are four buttons: 'Speichern' (Save), 'Speichern und schließen' (Save and close), 'Speichern und neu' (Save and new), and 'Speichern und zurück' (Save and back). The page footer includes 'Nach oben' (Back to top) and 'Frontend-Vorschau' (Frontend preview).

API-Schnittstelle

Auswahl der im System bereitgestellten Schnittstellen. Standard: *Standard-API*

Modus

Auswahl des Betriebsmodus, die mit der API bereitgestellt werden. Die Standard-API stellt den Modus "Export" und "Import" bereit.

The screenshot displays the 'API-Konfiguration' interface. It features two main sections: 'Cronjob-Einstellungen' and 'Ressourcen-Einstellungen'. In the 'Cronjob-Einstellungen' section, there is a checked checkbox for 'Cronjob aktivieren' with a subtext explaining its purpose. Below this is a dropdown menu for 'Cronjob' currently set to 'Wöchentlich'. The 'Ressourcen-Einstellungen' section includes a dropdown for 'Maximale Anzahl an Einträgen' set to '1'. At the bottom, there is an unchecked checkbox for 'Nur veröffentlicht'.

Cronjob-Einstellungen

☒ **Cronjob aktivieren**
Wählen Sie hier um einen Backend Cronjob für die Anbindung zu aktivieren.

Cronjob
Wöchentlich
Wählen Sie die Art des Cronjobs.

Ressourcen-Einstellungen

Maximale Anzahl an Einträgen
1
Wählen Sie die maximale Anzahl an Einträgen, die gemeinsam verarbeitet werden sollen. (Standard: 1). Alle = 0

☐ **Nur veröffentlicht**
Nur veröffentlichte Daten einbeziehen. Ein Veröffentlichungsfeld muss in der Tabellenkonfiguration angegeben sein.

Cronjob-Einstellungen

API Konfigurationen können mit Hilfe Contaos Cronjobs zeitgesteuert gestartet werden. Zur Auswahl steht:

- Wöchentlich
- Täglich
- Stündlich
- Jede Minute (minütlich)

Weitere Informationen zu Contao Cronjobs finden sich im [Contao Handbuch](#)

Ressourcen-Einstellungen

Maximale Anzahl an Einträgen

Legt fest wieviele Datensätze gleichzeitig bei der Ausführung der API-Schnittstelle abgearbeitet werden. Je nach Auswahl wird die Gesamtanzahl an Datensätzen in Pakete unterteilt.

Standard ist 1 Paket je Prozess. (langsamster Ablauf, ressourcen-schonend).

Einstellung "0" bzw. "-": Alle Pakete in einem Prozess (dies ist der schnellste, wenn auch ressourcen-aufwendigste Prozess)

2. API-Aufgaben (Jobs)

API-Aufgaben erlauben die Manipulation der Daten, die von Datenquelle zu Datenziel weitergereicht werden. Mit Hilfe von API-Jobs ist es zum Beispiel möglich Feldwerte im Laufe des API-Prozesses zu ändern, Konvertierungen von Daten durchzuführen oder ganze Felder auszuschließen.

API-Job Einstellung

The screenshot shows the 'API-Job Einstellung' form with several sections and annotations:

- Wahl des Aufgabentyps:** Points to the 'Aufgabentyp*' dropdown menu in the 'Allgemein' section, which is currently set to 'Feld-Aktion'.
- Zielfeld der Aufgabe, hier das Feld „id“:** Points to the 'Ziel*' dropdown menu in the 'Allgemein' section, which is currently set to 'id'.
- Aktions-Auswahl für Datenmanipulation:** Points to the 'Aktion' dropdown menu in the 'Wertermittlungs-Einstellungen' section, which is currently set to '-'.

The form includes the following sections and options:

- Allgemein:** Contains 'Aufgabentyp*' (Feld-Aktion) and 'Ziel*' (id).
- Einstellungen:** Contains a checkbox for 'Titel und Beschreibung eingeben'.
- Wertermittlungs-Einstellungen:** Contains 'Aktion' (set to '-') and 'Anwendungsregelung' (set to '-').
- Experten-Einstellungen:** Contains a checked checkbox for 'Aktivieren' and a 'Bei Fehler' dropdown (set to '-').

Buttons at the bottom: Speichern, Speichern und schließen, Speichern und neu, Speichern und zurück.

Aufgabentyp

Der Aufgabentyp definiert welcher Form der API-Job ist. Standardmäßig stehen Datenfeld-bezogene Aufgaben bereit.

Ziel

Die Ziel-Auswahl legt fest auf welches Datenfeld (bei Aufgabentype: Feld-Aktion) der Job angewendet wird. Die Auswahl richtet sich nach dem Datenziel in Abhängigkeit der API-Konfiguration (Bei Export z.B. die CSV-Datei. Bei Import das CC-Tabellenfeld)

Aktion

Legt die Art der Datenmanipulation fest und die Quelle der neuen Daten. Zur Auswahl stehen standardmäßig:

- Fixer Wert: Direkte Werteingabe möglich
- Ignorieren / ausschließen: Ziel wird in Datenziel ausgeschlossen
- Wert aus Datenquelle: Querkopie von Feld X nach Ziel aus der Datenquelle
- Eigene SQL-Abfrage: Wertermittlung aus einer eigenen SQL-Datenbankabfrage
- Hook-Callback-Funktion: Wertermittlung nach Ausführung einer Callback-Funktion
- Datumswert: Direkte Datumseingabe möglich
- Freitext: Eingabe von Fließtexten mit TinyMCE Editor
- Code / HTML: Eingabe von Codeblöcken
- Überschrift einsetzen: Direkte Werteingabe eines Überschriften-Werts

API-Prozesse ausführen

Die Ausführung der *Standard-API* über das Backend ist in 3 Phasen aufgeteilt:

I. Vorbereitungsphase:

Automatische Prozesse (je nach API-Konfiguration):

- Anlegen von Backups
- Leeren der aktuellen Datenbank

Beispiel: Vorbereitungsphase: Praxis-Beispiel: Export



II. Arbeitsphase:

Der API-Prozess wird gestartet... Die Quelldaten werden verarbeitet, durchlaufen den API-Aufgaben-Prozess (falls Aufgaben definiert sind), Zieldaten werden je nach API-Konfiguration abgearbeitet (z.B. in die Datenbank geschrieben bei einem Import). Abhängig der Ressourcen-Einstellungen wird der Prozess in einzelne Arbeitspakete geteilt und via Ajax weiterführend ausgeführt.

Jeder Arbeitsablauf wird protokolliert. Entstehende Fehler (z.B. SQL-Fehler oder php-Fehler) werden gesammelt.

Vollständig und fehlerfrei abgearbeitete Prozesspaket werden grün markiert. Treten während des Prozesses kritische Fehler auf, wird das Paket rot markiert.

Beispiel: Abgeschlossene Arbeitsphase: Praxis-Beispiel: Export



Hinweis: Dieser Prozess kann je nach Serverperformance und Arbeitsaufwand etwas Zeit in Anspruch nehmen.

III. Fertigstellung / Zusammenfassung

Nach Beendigung der Arbeitsphase werden gesammelte Statistika in Form eines Reports bereitgestellt.

Exporte werden geschrieben und bereitgestellt.

Beispiel: Zusammenfassung: Praxis-Beispiel: Export



Der API-Prozess ist damit abgeschlossen

Die *Standard-API*

Die Standard-API-Anbindung stellt zwei Betriebsmodi: Export und Import

Der Betriebsmodus Export steuert die Bereitstellung der CC-Daten an ein definiertes Ziel z.B. den Export der Daten in eine CSV-Datei.

Der Betriebsmodus Import steuert die Überführung von Fremddaten in die jeweilige CC-Tabelle. Die Datenquelle kann z.B. eine CSV-Datei sein oder eine Tabelle oder eine Datei bzw. dein Wertehaufen aus einer Hook-Funktion.

Die Ausführung von API-Jobs ist in allen Fällen optional. Die Weitergabe der Daten erfolgt dabei immer von Quelle -> Jobs -> Ziel. Im Falle eines Exports ist die Quelle die CC-Tabelle. Im Falle eines Imports ist die Quelle die jeweilige Datenquelle z.B. die CSV-Datei.

Ein API-Ablauf ohne Jobs entspricht einer direkten Überführung von Datenquelle/feld zu Datenziel/feld. (1-zu-1 Kopie)

[Praxis-Beispiel: Export](#)

[Praxis-Beispiel: Import](#)

Tutorials

Die folgenden Praxis-Beispiele basieren auf dem Tutorial "[Mein erster Katalog](#)".

Praxis-Beispiel: Export

Praxis-Beispiel: Import

Praxis-Beispiel: Export

Dieses Tutorial zeigt exemplarisch die Verwendung der *Standard-API*, Modus Export zum Exportieren bestehenden CC-Daten in eine CSV-Datei. Es wird die Konfiguration der API aufgezeigt und die Manipulation von Feldwerten mit Hilfe von API-Jobs. Im Anschluss wird der API-Prozess ausgeführt und die resultierende CSV-Datei in den Zielordner geschrieben.

[Schritt 1: API-Konfiguration](#)

[Schritt 2: Job-Konfiguration](#)

[Schritt 3: API ausführen](#)

Der Datensatz in "Mein erster Katalog" besteht aus 2 Datensätzen (hier: id=1, id=3).

The screenshot displays a web application interface for managing a catalog. On the left, a sidebar titled "Backend-Module" contains a menu with the following items: "Meine Kataloge" (expanded), "Mein erster Katalog" (selected), "Inhalte", "Layout", "Benutzerverwaltung", "System", and "Entwickler-Tools". The main content area is titled "Mein erster Katalog" and shows a list of two entries. The first entry, labeled "1", has a title "die_ueberschrift:" and a description "Eintrag 1". The second entry, labeled "3", has a title "die_ueberschrift:" and a description "Eintrag 2". Both entries include a text field and a text area. The interface also features a "Konfiguration bearbeiten" button, a "Neues Element" button, and a "Mehrere bearbeiten" button. At the bottom right, there are links for "Nach oben" and "Frontend-Vorschau".

Die Datensätze bestehen aus folgenden Feld-Attributen (Datenbank-Feldern)

Mein erster Katalog » Das Element 1 bearbeiten2.0.0

[Zurück](#)

Allgemein

Die Überschrift



























Eintrag 1h1

Ein Textfeld

Dies ist mein erster Eintrag

Eine Textarea

Datei Bearbeiten Einfügen Ansicht Format Tabelle



API-Konfiguration

Verweis: [Wo finde ich die API-Konfigurationen](#)

Schritt 1: Eine neue API-Anbindung hinzufügen

Wie gewohnt in Contao können neue Datensätze über die grünen "Plus"-Buttons angelegt werden.

Erstellen Sie einen neuen API Datensatz für den CC "Mein erster Katalog". Wählen Sie ggf. eine Einfügeposition für den neuen Datensatz. (Die Reihenfolge spielt keine Rolle. Jede API-Anbindung wird autark behandelt.)

☐


Schritt 2: Die API-Anbindung konfigurieren

Wählen Sie die *Standard-API*. Die Seite wird neuladen und die API-bezogenene Felder laden.

☐

Konfigurieren Sie die API nach folgendem Beispiel und speichern Sie den Datensatz.

Meine Inhaltselemente » Mein erster Katalog » Mein erster Katalog » API-Anbindung 1 bearbeiten 2.0.0



- ▾ API-Auswahl

API-Schnittstelle
Standard ▾
Wählen Sie eine der verfügbaren API-Schnittstellen aus
Die Standard-API-Schnittstelle stellt allgemeine Import-/ und Export Möglichkeiten bereit.

Modus
Export ▾
Wählen Sie den Modus dieser Anbindung

- ▾ API-Aufgaben-Einstellungen

Titel*
Mein Export
Geben Sie der Aufgabe einen Titel.

Beschreibung

Geben Sie der Aufgabe eine Beschreibung.

- ▾ Daten-Einstellungen

Ziel
CSV-Export ▾
Wählen Sie das Ziel der Datenübermittlung

Dateibezeichnung
meinExport_{{table}}
Geben Sie einen Dateinamen an. Inserttags sind erlaubt z.b.

Trennzeichen
Semikolon [;] ▾
Wählen Sie das Feld-Trennzeichen.

Zielpfad

Auswahl ändern

Wählen Sie den Zielpfad zum Speichern der Datei. (Standard: /files)

☐ **An Browser senden**
Die Datei nach dem Erstellen direkt zum Download bereitgeben

- ▷ Cronjob-Einstellungen

- ▷ Ressourcen-Einstellungen

- ▾ Experten-Einstellungen

☒ **Aktivieren**
Die API-Schnittstelle für die Verwendung bereitstellen.

Speichern

Speichern und schließen

Speichern und neu

Speichern und zurück

Kehren Sie zur API-Übersicht zurück -> "Speichern und schließen"

The screenshot shows a web interface for API configuration. The breadcrumb navigation at the top reads 'Meine Inhaltselemente » Mein erster Katalog » Mein erster Katalog'. The version number '2.0.0' is in the top right corner. Below the breadcrumb, there are filter controls: 'Filtern: API-Schnittstelle' and 'Modus'. A search bar is labeled 'Suchen: Titel' with a dropdown arrow and an equals sign. To the right of the search bar, it says 'Anzeigen: 1 - 1' with a dropdown arrow and a refresh icon. Below these controls, there are three buttons: 'Zurück' (with a left arrow), 'Neue API-Anbindung hinzufügen' (with a plus icon), and 'Mehrere bearbeiten' (with a group of people icon). The main content area shows the configuration for 'Mein erster Katalog'. It includes the 'Name' and 'Tabellenname' (cc_mein_erster_katalog). Below this, there is a section titled 'Mein Export' with a green plus icon and the text 'Mein Export'. At the bottom, it shows 'API-Schnittstelle: Standard' and 'Modus: Export'. On the right side of the 'Mein Export' section, there are several icons: a pencil, a green circle with a plus, a blue square with a plus, a green circle with a plus, a red X, and a clock.

Die API-Konfiguration ist fertig konfiguriert und kann ausgeführt werden. Optional können Jobs hinzugefügt werden.

[Weiter zu: Job-Konfiguration](#)

[Weiter zu: Ausführen](#)

Job-Konfiguration

API-Aufgaben bearbeiten.

Folgende Bedingungen soll der Export erfüllen:

- Der Feldwert des Feldes "ein_textfeld" soll geändert werden.
- Das Feld "id" soll nicht Bestandteil des Exports werden.

Vorschau

☐


Neue API-Aufgabe hinzufügen

Nutzen Sie die Funktion **Neue Aufgabe hinzufügen**, um einen neuen API-Aufgaben-Datensatz anzulegen. Wählen Sie ggf. die Einfügeposition. Die Reihenfolge der Aufgaben entspricht der Abarbeitungsreihenfolge während der API-Ausführung.

Job 1: Feld-Aktion: Fixer Wert

Konfigurieren Sie den API-Job nach folgendem Beispiel und speichern Sie den Datensatz.

Meine Inhaltselemente » Mein erster Katalog » Mein Export » Aufgabe 1 bearbeiten 2.0.0



Allgemein

Aufgabentyp*

Feld-Aktion

Wählen Sie die Art der Aufgabe

Ziel*

ein_textfeld

Wählen Sie das Ziel aus.

Einstellungen

☐ **Titel und Beschreibung eingeben**
Standardmäßig wird das Ziel der Aufgabe z.B. ein Feld als Titel genutzt

Wertermittlungs-Einstellungen

Aktion

Fixer Wert

Wählen Sie die auszuführende Aktion. Keine Auswahl = Leer

Festwert

Ich bin der neue Wert

Geben Sie einen festen Wert an. Order leer um das Feld leer zu

Experten-Einstellungen

☒ **Aktivieren**
Die Aufgabe für die Verwendung bereitstellen.

Anwendungsregelung

-

Erlaubt es diesen Job nur für bestimmte Anwendungsfälle

Bei Fehler

-

Wählen Sie welche Aktion bei einem Fehler ausgeführt werden

Speichern

Speichern und schließen

Speichern und neu


Speichern und bearbeiten

Kehren Sie zur API-Aufgaben-Übersicht zurück und ergänzen Sie einen neuen Datensatz oder wählen Sie "Speichern und neu" um einen weiteren API-Job anzulegen.

Job 2: Feld-Aktion: Ignorieren, ausschließen

Konfigurieren Sie den API-Job nach folgendem Beispiel und speichern Sie den Datensatz.

Meine Inhaltselemente » Mein erster Katalog » Mein Export » Aufgabe 2 bearbeiten 2.0.0

 Zurück

-▽ Allgemein

Aufgabentyp*

Feld-Aktion

Wählen Sie die Art der Aufgabe

Ziel*

pid

Wählen Sie das Ziel aus.

-▽ Einstellungen

☒ **Titel und Beschreibung eingeben**

Standardmäßig wird das Ziel der Aufgabe z.B. ein Feld als Titel genutzt

Titel*

PID ausschließen

Geben Sie der Aufgabe einen Titel.

Beschreibung

Das Feld PID soll nicht im Export erscheinen.

Geben Sie der Aufgabe eine Beschreibung.

-▽ Wertermittlungs-Einstellungen

Aktion

Ignorieren/ausschließen

Wählen Sie die auszuführende Aktion. Keine Auswahl = Leer

-▽ Experten-Einstellungen

☒ **Aktivieren**

Die Aufgabe für die Verwendung bereitstellen.

Anwendungsregelung

.

Erlaubt es diesen Job nur für bestimmte Anwendungsfälle

Bei Fehler

.

Wählen Sie welche Aktion bei einem Fehler ausgeführt werden

Speichern

Speichern und schließen

Speichern und neu

Speichern und zurück

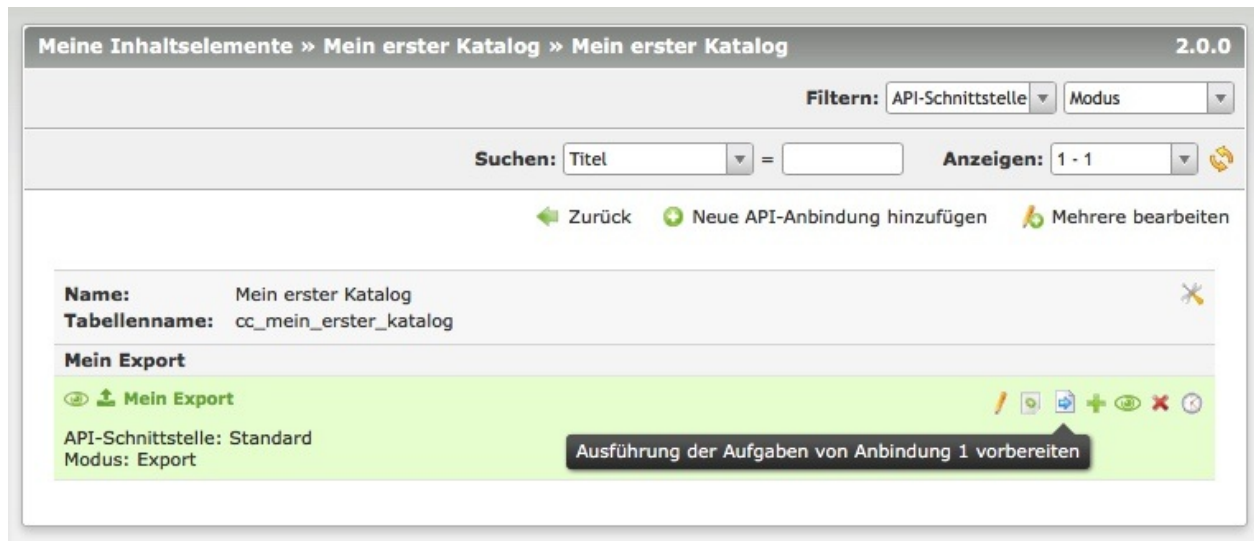
Die Konfiguration der API-Aufgaben ist damit abgeschlossen.

Prüfen Sie, dass die Übersicht der [Vorschau](#) in diesem Tutorial entspricht.

[Weiter zu: Ausführen](#)

Export ausführen

API-Prozess ausführen



1. Vorbereitungsphase

Die Ausführung kann gestartet werden. Automatische Prozesse sind abgeschlossen.

☐

Klicken Sie auf "Ausführen", um die Arbeitsphase zu starten.

2. Arbeitsphase

Die API verarbeitet die Daten... Dieser Prozess kann je nach Serverperformance und Arbeitsaufwand etwas Zeit in Anspruch nehmen.

☐

Klick Sie auf "Fertigstellen", wenn die Datenverarbeitung abgeschlossen ist.

3. Fertigstellen, Zusammenfassung

Die CSV-Datei wurde in Contaos /files Ordner geschrieben und steht zum Download bereit.

☐

Export-Datei überprüfen

files > meinExport_cc_mein_erster_katalog.csv

Öffnen Sie die Datei (z.B. mit Excel oder einem Texteditor) und überprüfen Sie den Export.

Ein Blick in die CSV Datei zeigt: Das "pid" Feld wurde entfernt und das Feld "ein_textfeld" wurde durch den fixen Wert ersetzt. Die anderen Feldwerte wurden direkt übertragen.

Die Feldnamen werden in Zeile 1 der CSV-Datei geschrieben.

Öffnen mit TextEdit								
id	tstamp	sorting	ptable	die_ueberschrift	ein_textfeld	eine_textarea	eine_select_auswahl	veroeffentlichen
1	1473154631	0		a:2:{s:4:unit;s:2:h1;s:5:value;s:9:Eintrag 1;}	Ich bin der neue Wert	<p>Lorem ipsum dolor sit amet, consi	wert1	1
3	1473154736	64		a:2:{s:4:unit;s:2:h1;s:5:value;s:9:Eintrag 2;}	Ich bin der neue Wert	<p>Auch gibt es niemanden, der den Schmerz ihm große Freude bereiten können.</p>	wert2	1

Tipp: Ergänzen Sie csv in dem Feld *Editierbare Dateien* in den Contao Systemeinstellungen, um die Datei mit Hilfe der Dateiverwaltung zu öffnen.

Dateien und Bilder

Erlaubte Download-Dateitypen

Hier können Sie eine kommagetrennte Liste von Dateitypen eingeben, die über

Editierbare Dateien

Hier können Sie eine kommagetrennte Liste von Dateitypen eingeben, die mit

Maximale Frontend-Breite

Wenn die Breite eines Bildes oder Films diesen Wert übersteigt, wird sie

Maximale GD-Bildbreite*

Hier können Sie die maximale Bildbreite eingeben, die die GD Library zu

Unterstützte Bildformate

Hier können Sie eine kommagetrennte Liste von Dateitypen eingeben, die von der

Unterstützte Templateformate

Hier können Sie eine kommagetrennte Liste von unterstützten Template-

JPG-Qualität

Hier können Sie die Qualität der JPG-Vorschaubilder in Prozent eingeben.

Maximale GD-Bildhöhe*

Hier können Sie die maximale Bildhöhe eingeben, die die GD Library zu

26

Praxis-Beispiel: Import

Dieses Tutorial zeigt exemplarisch die Verwendung der *Standard-API*, Modus Import, zum Importieren einer CSV-Datei. Es wird die Konfiguration der API aufgezeigt und die Manipulation von Feldwerten mit Hilfe von API-Jobs. Im Anschluss wird der API-Prozess ausgeführt.

Als Quelldatei dient die CSV-Datei aus dem [Praxis-Beispiel: Export](#).

Zusätzlich den bestehenden Einträgen wurden zwei weitere Einträge ergänzt, die als "neue" Einträge importiert werden sollen, wobei die bestehenden Einträge anhand der überschneidenden IDs in Form von Daten-Updates ausgeführt werden.

Ändern Sie die Quelldaten mit Hilfe eines Texteditors oder anderem CSV-fähigen Programm (Excel) (die bestehende CSV-formatierung muss erhalten bleiben) oder laden Sie die Beispiel-CSV-Datei. [Download](#) (Rechts-Klick: Link speichern unter...)

[Schritt 1: API-Konfiguration](#)

[Schritt 2: Job-Konfiguration](#)

[Schritt 3: API ausführen](#)

API-Konfiguration

Verweis: [Wo finde ich die API-Konfigurationen](#)

Hinweis: Die CSV-Quelldatei bezieht sich auf das [Praxis-Beispiel: Export](#)

Schritt 1: Eine neue API-Anbindung hinzufügen

Wie gewohnt in Contao können neue Datensätze über die grünen "Plus"-Buttons angelegt werden.

Erstellen Sie einen neuen API Datensatz für den CC "Mein erster Katalog". Wählen Sie ggf. eine Einfügeposition für den neuen Datensatz. (Die Reihenfolge spielt keine Rolle. Jede API-Anbindung wird autark behandelt.)



Schritt 2: Die API-Anbindung konfigurieren

Wählen Sie die *Standard-API*. Die Seite wird neuladen und die API-bezogenene Felder laden.



Konfigurieren Sie die API nach folgendem Beispiel und speichern Sie den Datensatz.

Meine Inhaltselemente » Mein erster Katalog » Mein erster Katalog » API-Anbindung 2 bearbeiten 2.0.0
Zurück

API-Auswahl

API-Schnittstelle
Standard

Wählen Sie eine der verfügbaren API-Schnittstellen aus

Modus
Import

Wählen Sie den Modus dieser Anbindung

Die Standard-API-Schnittstelle stellt allgemeine Import-/ und Export Möglichkeiten bereit.

API-Aufgaben-Einstellungen

Titel*
Mein Import

Geben Sie der Aufgabe einen Titel.

Beschreibung

Geben Sie der Aufgabe eine Beschreibung.

Daten-Einstellungen

Quelle
Datei

Wählen Sie die Quelle der Datenübermittlung

Trennzeichen
Semikolon [;]

Wählen Sie das Feld-Trennzeichen.

Dateiauswahl

files/meinExport_cc_mein_erster_katalog.csv (916,0 Byte)

Auswahl ändern

Wählen Sie die Datei aus

Update-Regeln-Einstellungen

Schlüsselfeld
id

Wählen Sie das einzigartige Schlüsselfeld dieser CC-Tabellen-

Schlüsselfeld in Quelle
id

Wählen Sie das einzigartige Schlüsselfeld der Quelle.

☒ **Einträge updaten**
Erlaubt das Überschreiben bestehender Datensätze anhand der

☐ **Einträge löschen**
Einträge löschen, die nicht in der Quelle existieren.

☒ **Neue Einträge anlegen**
Erlaubt das Anlegen neuer Einträge, wenn diese in der aktuellen Tabelle nicht existieren.

☐ **Tabelle vor Import leeren**
Leer die Tabelle bevor der Import ausgeführt wird.

Cronjob-Einstellungen

Backup-Einstellungen

☒ **Backup / Wiederherstellungspunkt anlegen**
Wählen Sie hier, wenn vor dem Ausführen ein Backup der aktuellen Tabelle angelegt werden soll. Das Backup wird in tl_undo erstellt.

Ressourcen-Einstellungen

Maximale Anzahl an Einträgen
1

Wählen Sie die maximale Anzahl an Einträgen, die gemeinsam verarbeitet werden sollen. (Standard: 1). Alle = 0

Experten-Einstellungen

☒ **Aktivieren**
Die API-Schnittstelle für die Verwendung bereitstellen.

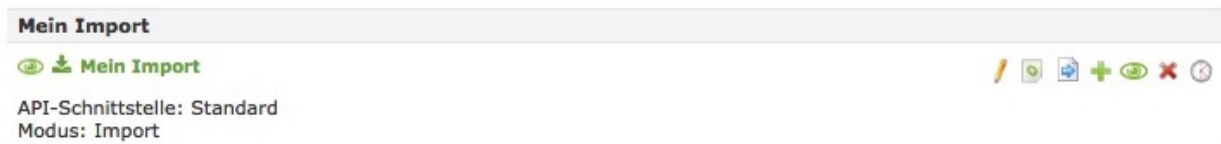
Speichern Speichern und schließen Speichern und neu Speichern und zurück

Update-Regel-Einstellungen

- Schlüsselfelder: Schlüsselfelder dienen der Identifizierung von überschneidenden Datensätzen. In diesem Beispiel hat sowohl die CC-Tabelle ein ID-Feld, als auch die Quell-Datei. Das ID-Feld ist in beiden Fällen ein eindeutiger Selektor eines Datensatzes und kann als Update-Schlüsselfeld genutzt werden.

Hinweis: Es müssen nicht identische Schlüsselfelder in Ziel und Quelle existieren. Lediglich der Wert in den Schlüsselfeldern dient als eindeutiger Selektor.

Kehren Sie zur API-Übersicht zurück -> "Speichern und schließen"



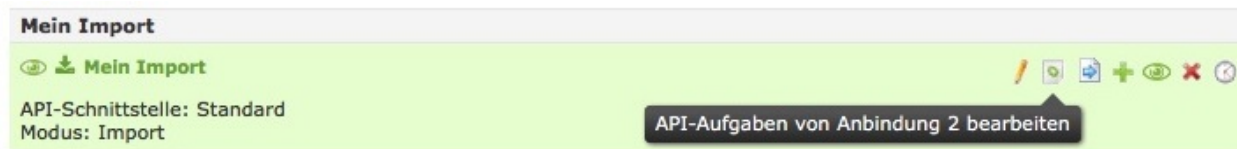
Die API-Konfiguration ist fertig konfiguriert und kann ausgeführt werden. Optional können Jobs hinzugefügt werden.

[Weiter zu: Job-Konfiguration](#)

[Weiter zu: Ausführen](#)

Job-Konfiguration

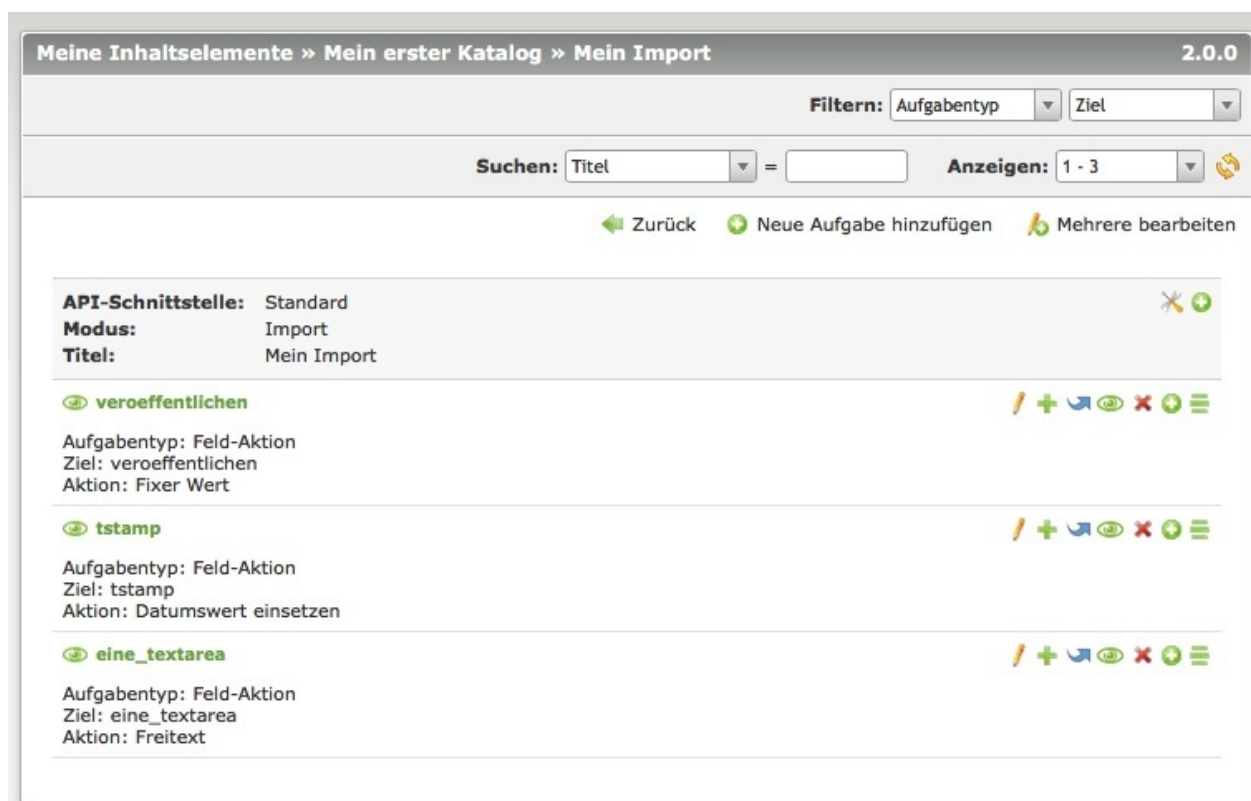
API-Aufgaben bearbeiten.



Folgende Bedingungen soll der Import erfüllen:

- Feldwert "veröffentlichen" soll "0" für neue Einträge (inserts) sein.
- Feldwert "tstamp" soll ein gewähltes Datum bekommen für neue Einträge.
- Neue Einträge sollen einen neuen Fließtext bekommen.

Vorschau



Neue API-Aufgabe hinzufügen

Nutzen Sie die Funktion **Neue Aufgabe hinzufügen**, um einen neuen API-Aufgaben-Datensatz anzulegen. Wählen Sie ggf. die Einfügeposition.

Die Reihenfolge der Aufgaben entspricht der Abarbeitungsreihenfolge während der API-Ausführung.

Job 1: Feld-Aktion: Fixer Wert

Konfigurieren Sie den API-Job nach folgendem Beispiel und speichern Sie den Datensatz.



Kehren Sie zur API-Aufgaben-Übersicht zurück und ergänzen Sie einen neuen Datensatz oder wählen Sie "Speichern und neu" um einen weiteren API-Job anzulegen.

Job 2: Feld-Aktion: Datumswert einsetzen

Konfigurieren Sie den API-Job nach folgendem Beispiel und speichern Sie den Datensatz.

Meine Inhaltselemente » Mein erster Katalog » Mein Import » Aufgabe 4 bearbeiten2.0.0

[Zurück](#)

Allgemein

Aufgabentyp*

Feld-Aktion

Wählen Sie die Art der Aufgabe

Ziel*

tstamp

Wählen Sie das Ziel aus.

Einstellungen

☐ **Titel und Beschreibung eingeben**
Standardmäßig wird das Ziel der Aufgabe z.B. ein Feld als Titel genutzt

Wertermittlungs-Einstellungen

Aktion

Datumswert einsetzen

Wählen Sie die auszuführende Aktion. Keine Auswahl = Leer

Datumswert

2016-09-15 13:44

Geben Sie oder wählen Sie das zu setzende Datum aus.

Experten-Einstellungen

☒ **Aktivieren**
Die Aufgabe für die Verwendung bereitstellen.

Anwendungsregelung

Nur für neue Datensätze (Inserts) anwenden

Erlaubt es diesen Job nur für bestimmte Anwendungsfälle

Bei Fehler

.

Wählen Sie welche Aktion bei einem Fehler ausgeführt werden

SpeichernSpeichern und schließenSpeichern und neuSpeichern und zurück

Nach obenFrontend-Vorschau

Tipp: Feld-Aktion: Fixer Wert mit dem Inserttag `{{date: :U}}` als Festwert erzeugt den aktuellen Zeitstempel.

Job 3: Feld-Aktion: Freitext

Konfigurieren Sie den API-Job nach folgendem Beispiel und speichern Sie den Datensatz.

Meine Inhaltselemente » Mein erster Katalog » Mein Import » Aufgabe 7 bearbeiten

2.0.0

Zurück

Allgemein

Aufgabentyp*

Feld-Aktion

Wählen Sie die Art der Aufgabe

Ziel*

eine_textarea

Wählen Sie das Ziel aus.

Einstellungen

☐

Titel und Beschreibung eingeben


Standardmäßig wird das Ziel der Aufgabe z.B. ein Feld als Titel genutzt

Wertermittlungs-Einstellungen

Aktion

Freitext

Wählen Sie die auszuführende Aktion. Keine Auswahl = Leer

Eigener Code 

Datei


Bearbeiten


Einfügen


Ansicht


Format


Tabelle










































































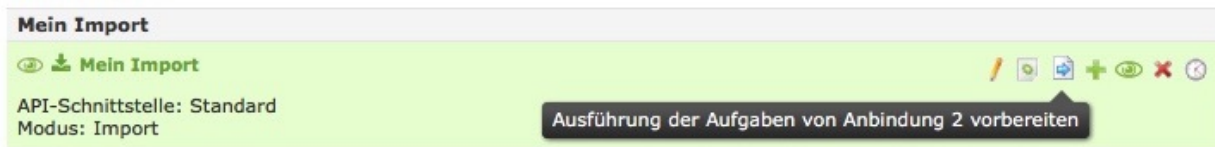
Die Konfiguration der API-Aufgaben ist damit abgeschlossen.

Prüfen Sie, dass die Übersicht der **Vorschau** in diesem Tutorial entspricht.

Weiter zu: Ausführen

Import ausführen

API-Prozess ausführen



1. Vorbereitungsphase

Die Ausführung kann gestartet werden. Automatische Prozesse sind abgeschlossen.

- Es wurde ein Backup der aktuellen Daten der Tabelle erstellt.



Klicken Sie auf "Ausführen", um die Arbeitsphase zu starten.

2. Arbeitsphase

Die API verarbeitet die Daten... Der Import-Prozess wird ausgeführt. Dieser Prozess kann je nach Serverperformance und Arbeitsaufwand etwas Zeit in Anspruch nehmen.

Meine Inhaltselemente » Aufgaben von Anbindung 2 ausführen2.0.0

 **API Anbindung 2 ausführen**

 **Import abgeschlossen.**

Data-Index: 0

Data-Index: 1

Data-Index: 2

Data-Index: 3


Zusammenfassung

Klick Sie auf "Fertigstellen", wenn die Datenverarbeitung abgeschlossen ist.

3. Fertigstellen, Zusammenfassung

Meine Inhaltselemente »**2.0.0**

[← Zurück zur Übersicht](#)

 **Zusammenfassung API Anbindung 2**

Report:

Data-Index: 0
Startzeit: 16.09.2016 08:47

Update
cc_mein_erster_katalog: 1
Endzeit: 16.09.2016 08:47
Prozesszeit in ms: 0
Fertiggestellt: true

Data-Index: 1
Startzeit: 16.09.2016 08:47

Update
cc_mein_erster_katalog: 3
Endzeit: 16.09.2016 08:47
Prozesszeit in ms: 0
Fertiggestellt: true

Data-Index: 2
Startzeit: 16.09.2016 08:47

Insert
cc_mein_erster_katalog: 72
Endzeit: 16.09.2016 08:47
Prozesszeit in ms: 0
Fertiggestellt: true

Data-Index: 3
Startzeit: 16.09.2016 08:47

Insert
cc_mein_erster_katalog: 73
Endzeit: 16.09.2016 08:47
Prozesszeit in ms: 0
Fertiggestellt: true

Nach oben Frontend-Vorschau

Hinweis: Es wurden zwei neue Datensätze angelegt: id=72, id=73

Daten überprüfen

Öffnen Sie die Daten des Katalogs.

Daten von „mein_erster_katalog“ nach erfolgreichem Import

Mein erster Katalog 2.0.0

Anzeigen: 1 - 4

Konfiguration bearbeiten Neues Element Mehrere bearbeiten

1
die_ueberschrift:
Eintrag 1
; ein_textfeld: Ich wurde geupdated; eine_textarea:
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor.
Aenean massa.

3
die_ueberschrift:
Eintrag 2
; ein_textfeld: Ich wurde geupdated; eine_textarea:
Auch gibt es niemanden, der den Schmerz an sich liebt, sucht oder wünscht, nur, weil er Schmerz
ist, es sei denn, es kommt zu zufälligen Umständen, in denen Mühen und Schmerz ihm große
Freude bereiten können.

7
die_ueberschrift:
Eintrag 3
; ein_textfeld: Ich bin Eintrag 3; eine_textarea:
Ich wurde gerade importiert.

Eintrag 4
; ein_textfeld: Ich bin Eintrag 4; eine_textarea:
Ich wurde gerade importiert.

Bestehende Daten
geupdated nach
CSV-Datei

Neue Datenangelegt:
- Wert in ein_textfeld nach CSV
- Wert in eine_textarea nach Job
- Nicht veröffentlicht

Wert in tstamp in der Datenbank
nach Datumswert belegt

Hinweis: Diese Darstellung ist nach dem Feld (id) sortiert (Standard). Die neuen Datensätze (id=72, id=73) sind daher in Bereich "7".

CustomCatalog API-Starter Frontend-Modul

Das CustomCatalog API-Starter Modul erlaubt API-Prozesse direkte aus dem Frontend heraus zu starten oder per direkter Adresse, was z. B. für automatisierte API-Prozesse mit Hilfe von realen serverseitigen cronjobs nötig ist.

The screenshot shows the Contao CMS 3.5.16 backend interface. The top navigation bar includes the user 'Benutzer admin', a 'Frontend-Vorschau' button, a 'Startseite' button, and an 'Abmelden' button. The left sidebar contains a 'Backend-Module' menu with categories like 'Meine Kataloge', 'Inhalte', 'Layout', 'Benutzerverwaltung', 'System', and 'Entwickler-Tools'. The main content area is titled 'Themes » Mein Theme für CustomCatalog » Frontend-Module » Modul ID 7 bearbeiten'. It features a version dropdown set to 'Version 7 (2016-09-13 15:26) admin' and a 'Wiederherstellen' button. The configuration is organized into sections: 'Titel und Typ' with fields for 'Titel*' (Mein CustomCatalog-API-Starter: Export) and 'Überschrift' (h1), and a 'Modultyp' dropdown set to 'CustomCatalog-API-Starter'; 'Modul-Konfiguration' with a 'Name*' dropdown set to 'Mein erster Katalog'; 'API-Einstellungen' with an 'API-Anbindung' dropdown set to 'Mein Export'; and 'Template-Einstellungen' with a 'Modul-Template' dropdown set to 'mod_customcatalogapi'. At the bottom, there are buttons for 'Speichern', 'Speichern und schließen', 'Speichern und neu', and 'Speichern und zurück'. A 'Zurück' button is also present at the top right of the configuration area.

Das Modul reagiert auf die GET-Parameter:

- api (ID-DER-API)
- run (1 = starten, 0 = nicht starten)
- limit (maximale Anzahl an Daten)
- offset (oberes Limit)

Eine exemplarische url, auf der das Modul platziert ist:

```
meineSeite.html?api=5&run=1&limit=10&offset=10
```

Lesbar: Starte API id=5, maximal 10 Daten der Quelle, Daten-Offset 10 (entspricht Quelldaten 10-20)

Developer

This section provides information about extending, modifying and manipulating the general API processes and gives an insight into the accessible hooks. The "MyAPI" tutorial will lead you through the general steps of creating your own custom API interface.

Since the source code is uncompiled, you can always take a look at the source code to get an even deeper understanding of how things are done.

Hooks

Reference

[Contao Cookbook](#)

CustomCatalog API Hooks

- [dataOutput](#)
- [apiComplete](#)

dataOutput

This hook is triggered after completing jobs and before passing the data to the output departments.

Available from CustomCatalog 2.0

Parameters

1. *array* `$arrSet`

The result set after completing the api jobs.

2. *object* (API) `$objAPI`

The api instance

Return values

Expects value `$arrSet` as return value or a sequential array `index => array(key => value)` that should be the output.

Example

```
<?php

// config.php
$GLOBALS['CUSTOMCATALOG_HOOKS']['dataOutput'][] = array('MyClass', 'myDataOutputCallba
ck');

// MyClass.php
public function myDataOutputCallback($arrSet, $strTable, $objApi)
{
    if($objApi->getTable() == 'my_cc_table')
    {
        foreach($arrSet as $i => $data)
        {
            // set pid to 10
            $data['pid'] = 10;

            // set data
            $arrSet[$i] = $data;
        }
    }

    return $arrSet;
}
```

apiComplete

This hook is triggered completing the api process.

Available from CustomCatalog 2.0

Parameters

1. *object* (API) `$objAPI`

The api instance

Example

```
<?php

// config.php
$GLOBALS['CUSTOMCATALOG_HOOKS']['apiComplete'][] = array('MyClass', 'myApiCompleteCall
back');

// MyClass.php
public function myApiCompleteCallback($objApi)
{
    if($objApi->getTable() == 'my_cc_table')
    {
        // api completed. do something
    }
}
```

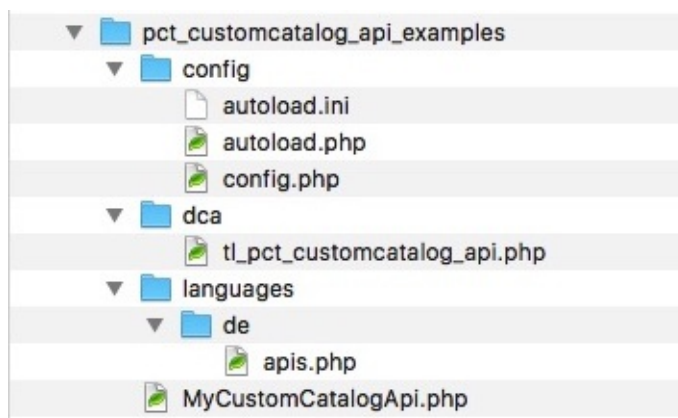
"MyAPI" Tutorial

This section will lead you through the basics of creating your own API interface. Knowledge of creating modules for Contao (version 3) are required as well as general knowledge of object oriented programming in PHP and webserver management (if you work online).

File structure

```
/system/modules/...
```

In this example, I use the name "pct_customcatalog_api_examples" as the name of my module folder.



I work in a German Contao backend environment, so I work with German (de) language files in this tutorial.

Register a new API

API configurations can be registered by creating a new node in the global

`$GLOBALS['PCT_CUSTOMCATALOG']['API']` array. Each API should have a unique key.

`.../config/config.php`

```
<?php

/**
 * Register new API
 */
$GLOBALS['PCT_CUSTOMCATALOG']['API']['myApi'] = array
(
    'label' => &$GLOBALS['TL_LANG']['PCT_CUSTOMCATALOG']['API']['myApi'],
    'class' => 'MyCustomCatalogApi', // The default class if no callbacks for modes ha
ve been registered
);
```

Now the API will be available in the [API selection](#).

Options [key]

- *array* "label"
Translations array (0=>title, 1=>description)
- *string* "class"
The main API php class
- *string* "path"
Path to a submodule folder nested inside a basic Contao module structure
- *array* "modes"
Register custom API modes
- *array* "key"
Registration array for backend pages (available keys: *run*, *summary*)


```
array('run'=> array(Myclass, MyCallback) );
```

* Register classes

Register the classes using Contao's autoloader.

```
.../config/autoload.php
```

```
<?php

/**
 * Register the classes
 */
ClassLoader::addClasses(array
(
    'MyCustomCatalogApi' => 'system/modules/pct_customcatalog_api_examples/MyCustomCatalogApi.php'
));
```

* Autoloading requirements

Since the API relies on the CustomCatalog extension

`pct_customelements_plugin_customcatalog` , tell Contao to load CC before this extension (requirement).

```
.../config/autoload.ini
```

```
;;
; List modules which are required to be loaded beforehand
;;
requires[] = "core"
requires[] = "pct_customelements_plugin_customcatalog"

;;
; Configure what you want the autoload creator to register
;;
register_namespaces = true
register_classes = true
register_templates = true
```

*This step can be done with the Contao Autoload-Creator once the file structure exists and class files contain valid php classes.

API configuration

Reference

[Contao DataContainer Array](#)

Table

`tl_pct_customcatalog_api`

Take a closer look at the table structure of the `tl_pct_customcatalog_api` table for a complete list of the fields available by default (reusability) before adding new fields to the table.

DataContainer array

`.../dca/tl_pct_customcatalog_api`

```
<?php

/**
 * Palettes
 */
$GLOBALS['TL_DCA']['tl_pct_customcatalog_api']['palettes']['myApi'] = '{title_legend},
type,be_description;{settings_legend},title,description;{backup_legend:hide},backup;{e
xpert_legend},published';
```

Meine Inhaltselemente » CC API » CC API Tester » API-Anbindung 26 bearbeiten
2.0.0

Zurück

API-Auswahl

API-Schnittstelle
Meine API
Wählen Sie eine der verfügbaren API-Schnittstellen aus
Dies ist meine eigene API-Schnittstelle.

API-Aufgaben-Einstellungen

Titel*
Meine API
Geben Sie dem API-Datensatz einen Titel.

Beschreibung
Dies ist meine eigene API-Schnittstelle
Geben Sie dem API-Datensatz eine Beschreibung.

Backup-Einstellungen

☒ **Backup / Wiederherstellungspunkt anlegen**
Wählen Sie hier, wenn vor dem Ausführen ein Backup der aktuellen Tabelle angelegt werden soll. Das Backup wird in tl_undo erstellt.

Experten-Einstellungen

☒ **Aktivieren**
Die API-Schnittstelle für die Verwendung bereitstellen.

Speichern
Speichern und schließen
Speichern und neu
Speichern und zurück

Available subpalette structure

By default the current palettes relate to the `tl_pct_customcatalog_api.type` selector field (the API selector). CustomCatalog also allows more complex selector-to-palette structures for selector fields:

- type
- mode

Example: Palettes for the *Standard* API: `type="standard", mode="import"`

```
'type_standard_mode_import' => '{title_legend},type,mode,be_description;{settings_legend},title,description;...'
```

Language files

```
.../languages/de/apis.php
```

```
<?php

/**
 * CustomCatalog API translations
 */
$GLOBALS['TL_LANG']['PCT_CUSTOMCATALOG']['API']['myApi'][0] = 'Meine API';
$GLOBALS['TL_LANG']['PCT_CUSTOMCATALOG']['API']['myApi'][1] = 'Dies ist meine eigene API-Schnittstelle.';
```

We use php files in this example. Feel free to use transifex files.

"MyCustomCatalogApi" class

.../MyCustomCatalogApi.php

```
<?php

/**
 * Class MyCustomCatalogApi
 */
class MyCustomCatalogApi extends \PCT\CustomCatalog\API\Base
{
    /**
     * Run
     * {@inheritdoc}
     * (is mandatory by interface)
     */
    public function run()
    {
        print('Executed API: '.$this->id);

        // create a backup record
        if($this->backup)
        {
            $this->createBackup();
        }
    }
}
```

The public function `run()` is mandatory for any API class that extends the `\PCT\CustomCatalog\API\Base` class.

It is recommended to extend the `\PCT\CustomCatalog\API\Base` class for any main API class. The base class provides many prebuilt methods to rely on when building your own api.

Reference

- class `\PCT\CustomCatalog\API\Base`
`pct_customelements_plugin_customcatalog/PCT/CustomCatalog/API/Base.php`
- interface `\PCT\CustomCatalog\Interfaces\ApiInterface`
`pct_customelements_plugin_customcatalog/PCT/CustomCatalog/Interfaces/ApiInterface.php`

Test your work

The front end should be your starting point for any custom-build APIs. The [CustomCatalog-API-Starter](#) module is the fastest way to develop your code when running your extension because you get direct feedback from your code without having to worry about any Contao back end requirements or additional programming besides your API logic. Once your core logic is developed, you can move on to build a back end execution.

Front end module

Create a new CustomCatalog-API-Starter module record and place the module in an article inside your page structure (just like any other Contao module).

Themes » default » Frontend-Module » Modul ID 59 bearbeiten

Version 2 (2016-09-21 09:00) admin Wiederherstellen

[Zurück](#)

Titel und Typ

Titel*

Bitte geben Sie den Titel des Moduls ein.

Überschrift

Hier können Sie dem Modul eine Überschrift hinzufügen.

Modultyp

Bitte wählen Sie den Typ des Moduls.

Modul-Konfiguration

Name*

Bitte wählen Sie die Konfiguration aus.

API-Einstellungen

API-Anbindung

Bitte wählen Sie die API-Anbindung, die gestartet werden soll.

Template-Einstellungen

Modul-Template

Wählen Sie ein individuelles Template für die Ausgabe des

Experten-Einstellungen

Front end output

Template: `mod_customcatalogapi.html5`

Switch to the front end and navigate to the page containing the module. The default visible output is just a regular submit button.

API Starter: Meine API

Ausführen

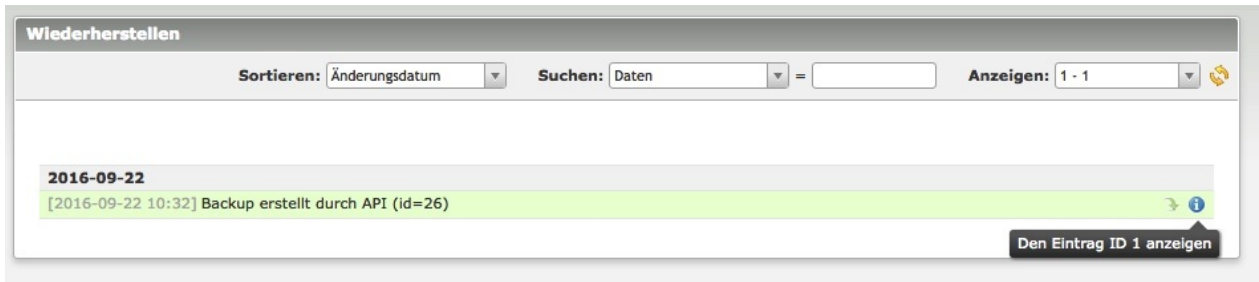
Execute

Click the submit button to execute the API process.

The API print the following information to the screen:

Executed API: 26
Backup created in tl_undo.id=1

A backup record has been created by API id=26



Ids might change due to your table autoincrement values

Happy coding...