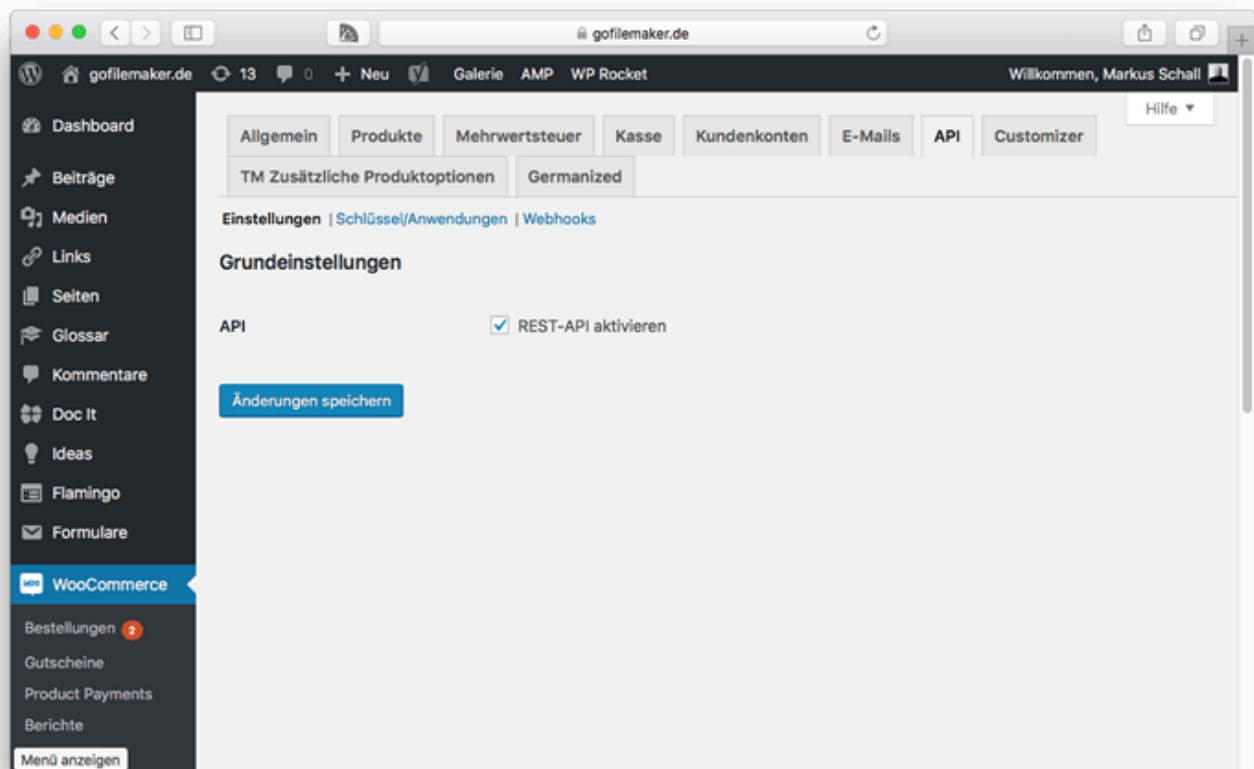


Bestellungen aus WooCommerce in FileMaker importieren

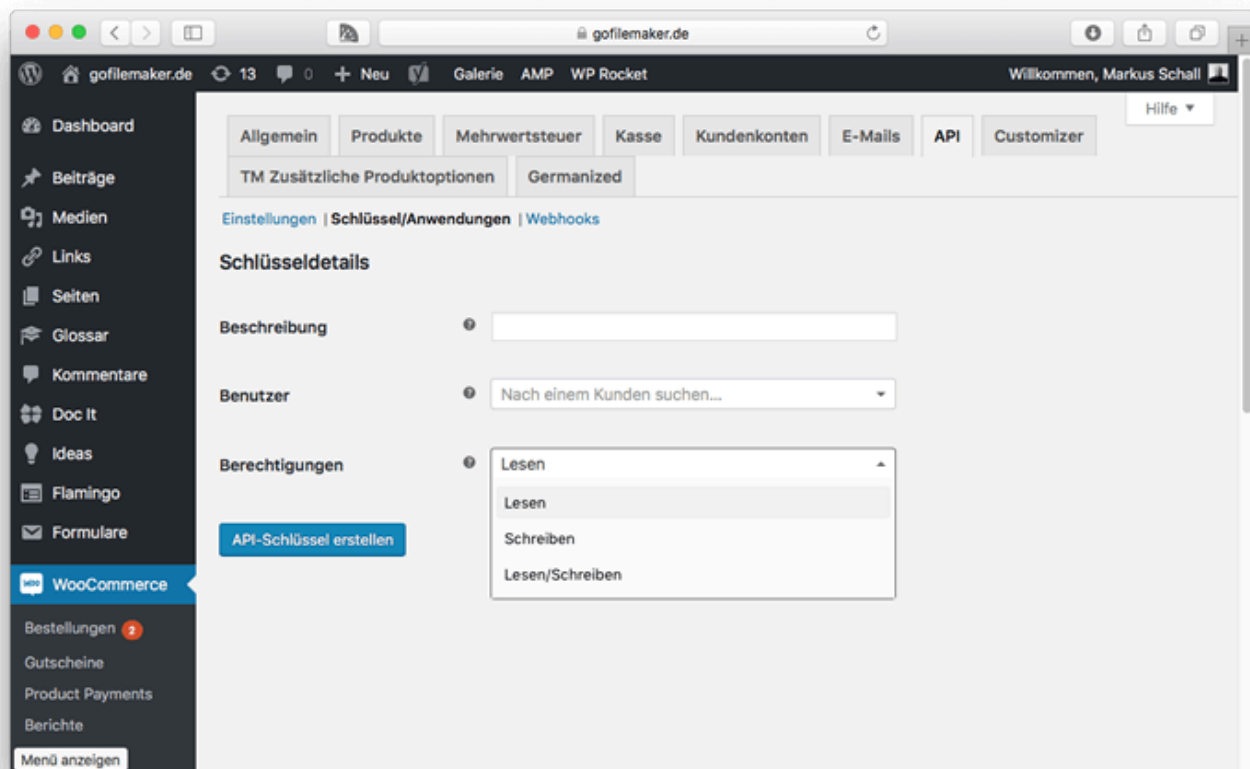
Der WooCommerce Onlineshop erfreut sich nicht zuletzt Dank der enorm hohen Verbreitung von WordPress-basierten Webseiten immer größerer Beliebtheit. Für die WooCommerce-Plattform sind heute unzählige Plugins zur Erweiterung verfügbar, und es ist heute kein Problem mehr, WooCommerce für Deutschland rechtssicher einzurichten. WooCommerce verfügt, wie viele andere Onlineshop- und Websysteme ebenfalls, über eine RESTful API, über die auf Daten des Onlineshops zugegriffen werden kann. In diesem Beitrag stellen wir eine Anleitung mit einer kleinen FileMaker-Lösung vor, mit der Bestellungen aus einem WooCommerce Onlineshop in eine FileMaker-[Datenbank](#) importiert werden können. Die Lösung bedient sich dabei der WooCommerce RESTful API v3 und wandelt den gelieferten JSON-Datenstrom in für FileMaker lesbare Daten um. Um die Programmierung der [Schnittstelle](#) zu erleichtern, wird für die Anmeldung an der RESTful API und für das Auslesen der JSON-Daten das MBS [FileMaker Plugin](#) von Monkeybread Software verwendet. Die Beispieldatenbank ist exemplarisch und kann prinzipiell auch für andere Onlineshopsysteme wie z.B. [Shopware](#), Magento oder [Gambio](#) GX3 eingesetzt werden.

Anmeldung an der WooCommerce RESTful API

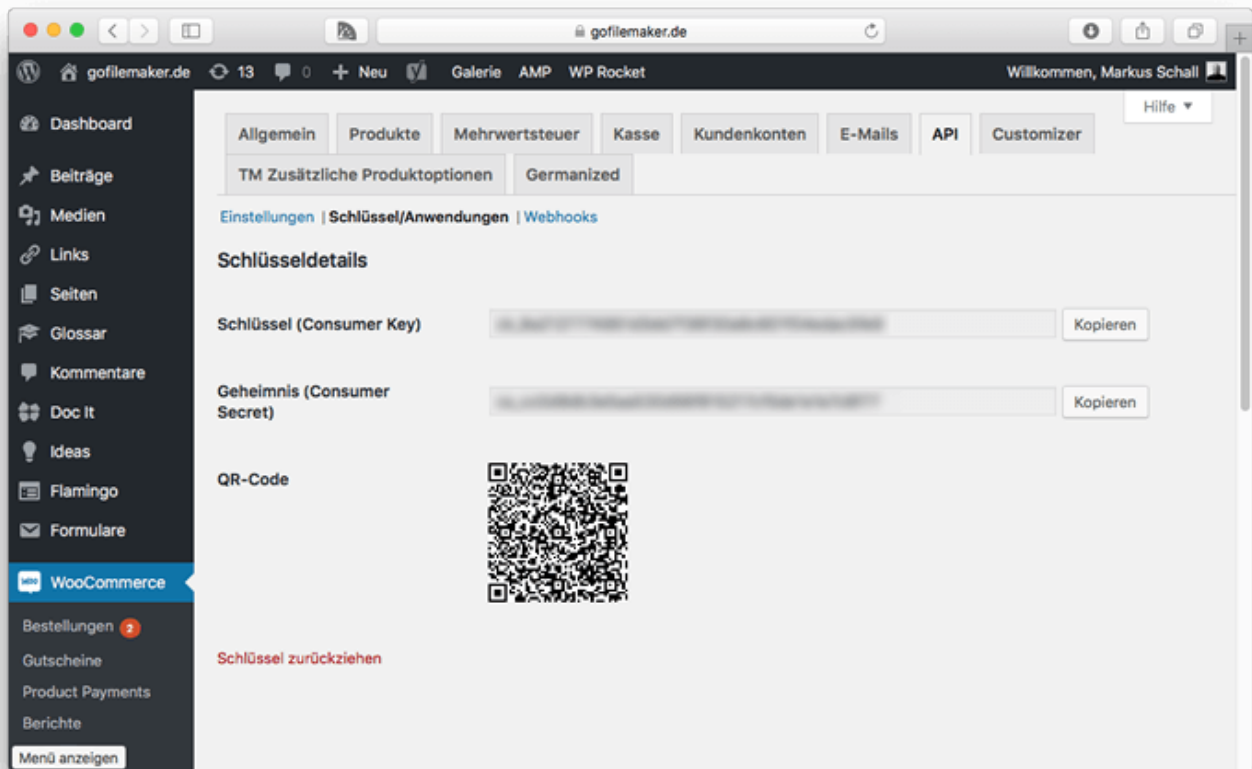
Um Daten von einem WooCommerce-Onlineshop auslesen zu können, muß zunächst die RESTful API in WooCommerce aktiviert werden. Diese Einstellung kann im WordPress-Backend unter *[WooCommerce > Einstellungen]* im Register *[API]* vorgenommen werden.



Klicken Sie auf die Schaltfläche *[Änderungen speichern]*, um die Aktivierung der [Schnittstelle](#) zu bestätigen. Im nächsten Schritt erzeugen Sie einen neuen API-Schlüssel. Klicken Sie dazu auf das Register *[Schlüssel/Anwendungen]* und daraufhin auf die Schaltfläche *[Schlüssel hinzufügen]*.



Im Feld *[Beschreibung]* können Sie eine beliebige Beschreibung hinzufügen. Wählen Sie einen Benutzer aus, dem der Zugriff auf die WooCommerce RESTful API gestattet werden soll. Je nach dem, ob der Benutzer auch Daten über die [Schnittstelle](#) in das Shopsystem schreiben soll, können Sie entsprechende Berechtigungen vergeben. Für die vorgestellte FileMaker-Lösung genügt die Berechtigung „Lesen“. Klicken Sie daraufhin auf die Schaltfläche *[API-Schlüssel erstellen]*.



Notieren Sie sich nun den Inhalt der beiden Felder „Schlüssel (Consumer Key)“ und „Geheimnis (Consumer Secret)“. Dieser Schritt ist wichtig, denn Sie können den Inhalt dieser Felder in Zukunft nicht mehr einsehen, da die Schlüsseldaten maskiert gespeichert werden.

Schnittstelle in FileMaker zur WooCommerce REST API

Mit der in diesem Beitrag vorgestellten Beispiellösung können Shopbestellungen und verbundene Kundendaten wie Rechnungs- und Lieferanschrift aus einem WooCommerce Onlineshop in eine FileMaker-[Datenbank](#) übertragen werden. Die Beispieldatei (oder eigene Lösung) kann problemlos um weitere Funktionen und Tabellen erweitert werden. Folgende Endpunkte stellt WooCommerce für den Zugriff auf die RESTful API zur Verfügung:

- Coupons
- Customers
- Orders (wird in der Beispieldatei genutzt)

- Order Notes
- Refunds
- Products
- Product attributes
- Product attribute terms
- Product categories
- Product shipping classes
- Product tags
- Reports
- Tax rates
- Tax classes

Mit der WooCommerce RESTful API können also auch noch viele weitere Datenquellen mit FileMaker ausgelesen und/oder bearbeitet werden wie beispielsweise Gutscheine, Produkte, Produktattribute oder auch Steuerklassen. In unserer FileMaker-Beispiellösung verwenden wir den Endpunkt „Orders“, um Bestellungen aus WooCommerce in FileMaker zu importieren.

Beispiellösung benötigt das MBS FileMaker Plugin

Für die Entwicklung der [Schnittstelle](#) wurde in der Beispieldatei das [MBS FileMaker Plugin](#) von Monkeybread Software verwendet. Das Plugin bietet alle Funktionen, die für eine [Schnittstelle](#) mit RESTful API erforderlich sind. Alternativ könnten auch Funktionen des kostenlos erhältlichen [Base Elements Plugin](#) genutzt werden, wenn die Software entsprechend umprogrammiert wird.

Voreinstellungen in der FileMaker-Lösung

Die Testversion für die FileMaker-Lösung der [Schnittstelle](#) zu einem WooCommerce Onlineshop können Sie [hier herunterladen](#).

Die Lösung liegt im [fmp12-Format](#) für FileMaker 13 bis 15 vor und benötigt das MBS [FileMaker Plugin](#) mindestens in Version 4.4. Nachdem Sie das Zip-Archiv entpackt und die Lösung mit FileMaker gestartet haben, müssen Sie zunächst die Daten für die RESTful-API Ihres WooCommerce Onlineshops eintragen.

RESTful API Zugangsdaten

Fremdsystem: ☐ SSL

API-URL:

Endpunkt:

Startobjekt:

Consumer-Key:

Consumer-Secret:

Rechnungsanschrift

Firma:

Vorname:

Name:

Anschrift:

Land/PLZ/Ort:

Lieferanschrift

Firma:

Vorname:

Name:

Anschrift:

Land/PLZ/Ort:

Stammdaten der Bestellung

_Bestellnummer: ID:

Datum_Bestellung:

Bestellstatus:

Zahlungsmethode:

Versandmethode:

Bemerkungen:

Währung:

Bestellpositionen

Anz.	Artikel-Nr.	Artikelbezeichnung	MwSt.	Betrag
1		gFM-Business Custom Basic Mac, Windows, iPad, Einzelplatz	37,81	199,00
1		gFM-Business Custom Basic Mac, Windows, iPad, Mehrplatz	85,12	448,00
1		gFM-Business Light für Mac, Windows und iPad, Einzelplatzlizenz	56,81	299,00

Datenausgabe JSON

```
{
  "orders": [
    {
      "id": "15285",
      "order_number": "15285",
      "order_key": "wc_order_584534e9da8b6",
      "created_at": "2016-12-05T09:35:37Z",
      "updated_at": "2016-12-05T09:35:37Z",
      "completed_at": "2016-12-05T09:35:37Z"
    }
  ]
}
```

Datenausgabe Bestellung

```
{
  "id": "15285",
  "order_number": "15285",
  "order_key": "wc_order_584534e9da8b6",
  "created_at": "2016-12-05T09:35:37Z"
}
```

Summe netto: **946,00**

MwSt.: **179,74**

Summe brutto: **1.125,74**

Datenimport

Tragen Sie alle Daten in die mit gelber Farbe hinterlegten Felder ein:

- **API-URL** – URL Ihres Servers und die Endung `/wp-api/v3`
- **Endpunkt** – Standard: `/orders`
- **Startobjekt** – Standard: `orders`
- **Consumer-Key** – Consumer-Key, den Sie im WordPress-

Backend erstellt und notiert haben

- **Consumer-Secret** – Consumer-Secret, das Sie im WordPress-Backend erstellt und notiert haben

Wenn alle Daten korrekt eingetragen wurden, werden nach Klick auf die Schaltfläche *[Datenimport]* alle Bestellungen des angegebenen WooCommerce Onlineshops und die zugehörigen Kundendaten importiert.

Aufbau der FileMaker Beispieldatenbank

Die Beispieldatenbank für den Import von Shopbestellungen ist einfach aufgebaut und enthält folgende Datentabellen:

- **Bestellungen** – Stammdatensatz einer Bestellung, enthält folgende Felder:
 - `__ORDERS` (Daten aus dem Startobjekt „orders“, dient nur zur Veranschaulichung)
 - `_Bestellnummer` (Bestellnummer aus WooCommerce)
 - `_ID_Fremdsystem` ([Primärschlüssel](#) *[post_id]* aus WooCommerce)
 - `_pk_Bestellung_ID` ([Primärschlüssel](#))
 - `Bemerkungen` (Bemerkungen aus WooCommerce)
 - `Bestellstatus`
 - `Betrag_Gesamt_brutto`
 - `Betrag_Gesamt_netto`
 - `Betrag_MwSt`
 - `Betrag_MwSt_Positionen`
 - `Betrag_MwSt_Versandkosten`
 - `Betrag_Rabatt`
 - `Betrag_Versandkosten`
 - `Betrag_Zwischensumme`
 - `Datum_Bestellung`
 - `E_Mail`
 - `JSON_Ausgabe` (Globales Feld für Ausgabe der JSON-

Daten, wird im [Script](#) verwendet)

- Kunden_Nr_WC
- LiAdr_Anschrift (Lieferanschrift)
- LiAdr_Firma
- LiAdr_Ländercode
- LiAdr_Name
- LiAdr_Ort
- LiAdr_PLZ
- LiAdr_Vorname
- Mehrwertsteuersatz
- ReAdr_Anschrift (Rechnungsanschrift)
- ReAdr_Firma
- ReAdr_Ländercode
- ReAdr_Name
- ReAdr_Ort
- ReAdr_PLZ
- ReAdr_Vorname
- Status_bezahlt
- Telefon
- Versandmethode
- Währung
- Zahlungsmethode

▪ **Positionen** – [Tabelle](#) von Bestellpositionen, enthält folgende Felder:

- _Bestellnummer (Bestellnummer aus WooCommerce)
- _pk_Position_ID ([Primärschlüssel](#))
- Anzahl
- Artikel_Nr
- Artikelbezeichnung
- Betrag_MwSt
- Betrag_netto
- id ([Primärschlüssel](#) aus WooCommerce)
- product_id (ID des Produktes in WooCommerce)

▪ **Einstellungen** – Einstellungen zur WooCommerce RESTful

API

- Fremdsystem (Global) – Name des Fremdsystems (Nur verwendet bei Einsatz mehrerer Fremdsysteme)
- rf_URL (Global) – URL des Servers mit Pfad zur API
- rf_Benutzer (Global) – Benutzername/Consumer Key für RESTful API
- rf_Passwort (Global) – Passwort/Consumer Secret
- rf_Status_SSL (Global) – Status, ob SSL verwendet werden soll oder nicht
- rf_Endpunkt (Global) – Endpunkt der RESTful API
- rf_Startobjekt (Global) – Startobjekt innerhalb des Endpunktes

Die Tabellen „Bestellungen“ und „Positionen“ sind über den Schlüssel „_Bestellnummer“ miteinander verknüpft. Für die Integration in eigene FileMaker-Lösungen können die Felder in diesen Tabellen beliebig umbenannt werden. Die [Tabelle](#) „Einstellungen“ enthält keine Verknüpfung, weil alle Felder global definiert wurden. Wenn mehrere Drittsysteme mit RESTful API angebunden werden sollen, ist es empfehlenswert, die Felder nicht als globale Felder zu definieren und einen [Primärschlüssel](#) zur Auswahl des Fremdsystems hinzuzufügen.

Anmeldung an der RESTful API

Die FileMaker Beispiellösung enthält zwei Teilscripte, mit denen die Bestelldaten aus dem Onlineshop in FileMaker importiert werden können. Im [Script](#) „ts.Query_REST_API“ erfolgt zunächst die Anmeldung am [Server](#). Dazu werden mit dem MBS-Plugin entsprechende cURL-Funktionen mit folgendem Ablauf aufgerufen:

- Neue cURL-Session öffnen
- URL für REST-API setzen
- Logindaten setzen

- Header-Option setzen
- SSL setzen
- Log-Messages einschalten
- Abfrage ausführen mit Meldung bei Fehler
- Ergebnis in das Feld `[Bestellungen::JSON_Ausgabe]` schreiben
- cURL-Session schließen, Cleanup

```

1 # Neue CURL-Session
2 Variable setzen [$curl; Wert:MBS("CURL.New")]
3 # URL für REST-API setzen
4 Variable setzen [$r; Wert:MBS("CURL.SetOptionURL"; $curl; Einstellungen::rf_URL & Einstellungen::rf_Endpunkt)]
5 # Logindaten setzen
6 Variable setzen [$u; Wert:MBS("CURL.SetOptionUsername"; $curl; Einstellungen::rf_Benutzer)]
7 Variable setzen [$p; Wert:MBS("CURL.SetOptionPassword"; $curl; Einstellungen::rf_Passwort)]
8 Variable setzen [$a; Wert:MBS("CURL.SetOptionHTTPOAuth"; $curl; 15 )]
9 # Header-Option setzen
10 Variable setzen [$h; Wert:MBS("CURL.SetOptionHTTPHeader"; $curl; "Content-Type: application/json"; "Expect:" )]
11 # SSL setzen
12 Variable setzen [$s; Wert:MBS("CURL.SetOptionSSLVerifyHost"; $curl; Wenn(IstLeer(Einstellungen::rf_Status_SSL); 0; 1) )]
13 Variable setzen [$v; Wert:MBS("CURL.SetOptionSSLVerifyPeer"; $curl; 0 )]
14 # Log-Messages einschalten
15 Variable setzen [$l; Wert:MBS("CURL.SetOptionVerbose"; $curl; 1 )]
16 # Abfrage ausführen
17 Variable setzen [$error; Wert:MBS("CURL.Perform"; $curl )]
18 # Fehlermeldungen
19 Variable setzen [$errormessage; Wert:MBS("CURL.GetDebugAsText"; $curl)]
20 Wenn [$error ≠ "OK"]
21   Eigenes Dialogfeld anzeigen ["Error"; $error & " | " & $errormessage]
22   Aktuelles Script verlassen []
23 Ende (wenn)
24 # Ergebnis
25 Variable setzen [$output; Wert:MBS("CURL.GetResultAsText"; $curl)]
26 Feldwert setzen [Bestellungen::JSON_Ausgabe; $output]
27 # Cleanup
28 Variable setzen [$c; Wert:MBS("CURL.Cleanup"; $curl )]

```

JSON-Daten der Schnittstelle auslesen

Mit dem zweiten Teilsript „ts.Parse_JSON_Bestellungen_WC“ werden die vom [Server](#) gelieferten JSON-Daten ausgelesen und in die einzelnen Felder der Tabellen „Bestellungen“ und „Positionen“ eingetragen. Dabei werden zunächst alle Daten aus dem Startobjekt „orders“ (siehe Feld „Startobjekt“ in den Einstellungen) in die [Variable](#) \$ordersJSON geschrieben. Darauf folgt eine Haupt-Ausführungsschleife zum Auslesen aller Bestellungen und eine weitere Ausführungsschleife für den Import aller Positionen zu einer Bestellung. Ein Zugriff auf den [Server](#) erfolgt in diesem [Script](#) nicht mehr.

```

1 Variable setzen [$json; Wert:Bestellungen::JSON_Ausgabe]
2 Variable setzen [$ordersJSON; Wert:MBS("JSON.GetObjectItem"; $json; Einstellungen::rf_Startobjekt)]
3 Variable setzen [$counter; Wert:0]
4 # Vorhandene Datensätze löschen
5 Alle Datensätze löschen [Ohne Dialogfeld]
6 Gehe zu Layout ["sys.Positionen" (Positionen)]
7 Alle Datensätze löschen [Ohne Dialogfeld]
8 Gehe zu Layout [Originallayout]
9 # Haupt-Ausführungsschleife
10 Schleife (Anfang)
11   Variable setzen [$order; Wert:MBS("JSON.GetArrayItem"; $ordersJSON; $counter)]
12   Verlasse Schleife wenn [IstLeer($order)]
13   Variable setzen [$counter; Wert:$counter + 1]
14   # STAMMDATENSATZ
15   Variable setzen [$ID_Fremdsystem; Wert:MBS("JSON.GetIntegerValue"; MBS("JSON.GetObjectItem"; $order; "id" ))]
16   Variable setzen [$Bestellnummer; Wert:MBS("JSON.GetIntegerValue"; MBS("JSON.GetObjectItem"; $order; "order_number" ))]
17   Variable setzen [$Datum_Bestellung; Wert:MBS("JSON.GetStringValue"; MBS("JSON.GetObjectItem"; $order; "created_at" ))]
18   (...)
51  # Stammdatensatz erzeugen
52  Neuer Datensatz/Abfrage
53  Feldwert setzen [Bestellungen::__ORDERS; $order]
54  Feldwert setzen [Bestellungen::__ID_Fremdsystem; $ID_Fremdsystem]
55  Feldwert setzen [Bestellungen::__Bestellnummer; $Bestellnummer]
56  Feldwert setzen [Bestellungen::__Datum_Bestellung; $Datum_Bestellung]
57  (...)
89  # BESTELLPOSITIONEN
90  Gehe zu Layout ["sys.Positionen" (Positionen)]
91  Variable setzen [$counter_pos; Wert:0]
92  Schleife (Anfang)
93    Variable setzen [$position_id; Wert:Austauschen(MBS("JSON.GetObjectItemTree"; $order; "line_items"; $counter_pos; "id");
94    Verlasse Schleife wenn [IstLeer($position_id)]
95    Variable setzen [$Anzahl; Wert:Austauschen(MBS("JSON.GetObjectItemTree"; $order; "line_items"; $counter_pos; "quantity"
96    (...)
101  # Neue Position erfassen
102  Neuer Datensatz/Abfrage
103  Feldwert setzen [Positionen::__Bestellnummer; $Bestellnummer]
104  Feldwert setzen [Positionen::__id; $position_id]
105  Feldwert setzen [Positionen::__Anzahl; $Anzahl]
106  (...)
111  Variable setzen [$counter_pos; Wert:$counter_pos + 1]
112  Schleife (Ende)
113  Gehe zu Layout [Originallayout]
114  Schleife (Ende)

```

Alle Felder der Lieferanschrift (shipping_address), Rechnungsanschrift (billing_address) und Belegpositionen (line_items) befinden sich in einer Baumstruktur und müssen mit dem Befehl: `MBS( „JSON.GetObjectItemTree“; json; names )`

aus dem entsprechenden Baum ausgelesen werden. Alle anderen Felder können ganz normal mit den Funktionen `MBS( „JSON.GetStringValue“; json )` oder `MBS( „JSON.GetIntegerValue“; json )` ausgelesen werden.

Download FileMaker Beispiel-Datenbank

Die FileMaker-Beispieldatenbank können Sie [hier herunterladen](#) und mit einem beliebigen WooCommerce Onlineshop testen. Die öffentliche Version ist für die Bearbeitung von Felddefinitionen und Scripten gesperrt. Sie möchten mit der Beispiellösung sofort starten und die Lösung an Ihre eigenen Anforderungen anpassen? Dann nehmen Sie gerne [Kontakt](#) mit uns auf.



FileMaker Schnittstelle

WooCommerce-Bestellungen per RESTful API in FileMaker importieren

Anleitung für den Import von Shopbestellungen aus einem WooCommerce Onlineshop in eine FileMaker-Datenbank über die WooCommerce RESTful API.