Abschlussprojekt 2022/2023

Dokumentation von Pierre Behlert

(im Folgenden als „Autor“ bezeichnet)

Praktikumsbetrieb:

Projektbezeichnung:

Konzeption und Erstellung einer Schnittstelle zwischen einem Warenwirtschaftssystem und zweier Onlineshops zur Artikelsynchronisation unter Einbeziehung einer Rest API

Bildungsträger:

Inhaltsverzeichnis

Verzeichnisse

Tabellenverzeichnis:

Grafikverzeichnis:

1 Einleitung

2 Das Projektumfeld

2.1 Bildungsträger

Der Bildungsträger, über den diese Umschulung läuft ist die „iad“. Der Name ausgeschrieben lautet „Informationsverarbeitung und angewandte Datentechnik GmbH“. Bei diesem Bildungsträger werden den Personen sämtliche Grundlagen beigebracht, die einem den Einstieg in die Informatikbranche erleichtern. Seit 39 Jahren sind Sie als Bildungspartner für mehrere Standorte und schult erfolgreich Personen zu spezifischen Berufen um. *(vgl. https://www.iad.de/)*

2.2 Praktikumsbetrieb

Die Firma „k-i-sys“ existiert seit über 20 Jahren und ist mit seinen Mitarbeitern in jeglichen Bereichen, rund um die Bereiche IT-Infrastruktur, Beratung, eCommerce, Entwicklung Web & App tätig. Des Weiteren ist der Support im Service enthalten. Das bedeutet, dass die meisten Kunden schon eine langjährige Bindung zu diesem Unternehmen haben. In Kooperation eines „Gelebten Netzwerks“ namens „kreativkollegen®“ sind Sie damit sehr erfolgreich.

(vgl. https://kisys.de/ & https://www.kreativkollegen.de/)

Der Autor hat dort sein Praktikum als Anwendungsentwickler gemacht und konnte dort auch in Bereichen rund um die Web-Entwicklung und den Kundensupport einiges an Erfahrung mitnehmen.

3 Das Projekt

3.1 Projektdefinition (Auftrag von Kunde)

Der Kunde benötigt für seine WordPress Seite folgende Funktionalitäten, um seine Verkaufszahlen und Sortiment zu erweitern (u.a. wegen Kooperation mit anderem Online-Shop):

Bestellungen	Wenn Bestellung „eingeht“, diese in eine CSV-Datei schreiben, nach 24 Stunden den Status via. Cronjob ändern – Vorbereitungen für separaten Eintrag zum Beispiel für die Warenwirtschaft erstellt
User/Kunden	User/Kunden, wenn Sie sich registrieren oder etwas bestellen in eine CSV-Datei einpflegen (auch unregistrierte Benutzer)
Rest API	Produktsynchronisation zwischen 2 Online-Shops

Tab. 1: WordPress Funktionalitäten

3.2 Make-or-Buy-Entscheidung

Grundlegend gibt es die Möglichkeit, andere Applikationen wie z.B. WP-ALL-IMPORT/EXPORT für das rausschreiben der Bestellungen und User zu nutzen, aber für die Rest API gibt es im Internet kein alternatives Plugin. Die Rest API ist explizit darauf ausgelegt, eine händische Alternative zu erstellen.

Dementsprechend kam der Autor zu dem Entschluss, den Auftrag komplett als eigenes Plugin zu erstellen und für eventuell weitere Aufträge, die eine Rest API nutzen wollen eine vorgefertigte Vorlage zu besitzen.

3.3 Projektplanung

Projektphasen mit Zeitplanung in Stunden	Stunden
Analysephase	
Kundenbesprechung und Vorhaben-Analyse	2
Zieldefinition	2
Konzeption	8
Abstimmung intern und mit Kunden, Auftrag	2
Realisierung	
Schnittstelle Shop ←→WW (Warenwirtschaft)	33
Synchronisation Shop 2 ←→Shop 1	22
Abschluss	
Testing und Übergabe	4
Dokumentation	7
Gesamtzeit	**80**

Tab. 2: Zeitliche Planung des Projektes

4 Technische Daten (Hard- & Software)

Der Server, auf dem beide Webshops laufen ist ein gehärteter Gentoo-Server.

Es wird die neueste Version von WordPress & WooCommerce, ebenfalls wird die Programmiersprache PHP in der Version 7.4 genutzt.

Als Software hat der Autor zu Beginn Visual Studio Code benutzt, bis er dann aufgrund mehrerer Gründe auf PHPStorm gewechselt hat. Gründe dafür sind, dass firmenintern serverseitige Windowsprofile genutzt werden und Visual Studio Code den Cache überfüllt hat durch seine ganzen Plugins. Zu Testzwecken hat er sich für Microsoft Edge die Erweiterung namens Advanced API Client benutzt, ebenso auch meistens Microsoft Edge oder Mozilla Firefox als Browser.

Des Weiteren wurde während der Entwicklung GitHub genutzt, um die verschiedenen Versionsunterschiede und Änderungen sehen zu können. Somit war es ein leichtes, die alte Version wiederherzustellen, wenn man einen Fehler gemacht hat und die Seite zum Beispiel einen kritischen Fehler anzeigt.

5 Kundenanforderungen

Die erste Anforderung ist, dass sowohl Bestellungen, als auch Registrierungen auf der Seite wahrgenommen werden und in eine lesbare Datei geschrieben werden sollen. Format wurde dem Autor mit einer kleinen Auswahl freigestellt.

Anforderung Nummer 2 ist eine Artikelgruppe aus einem Partnershop (Kunde B) ebenfalls in dem Shop von Kunde A einbinden mit einem Abgleich der Daten.

Die Alternative wäre, dass ein Mitarbeiter des Kunden regelmäßig händisch die Datensätze überprüft und pflegt.

5.1 Amortisationsrechnung

Beschreibung	Zeit	€/Std	Gesamtpreis
Ausgaben Auszubildender	70 Stunden á 12€	12€	840€

Tab. 3: Kosten Autor

Beschreibung	Zeit	€/Std	Preis/Jahr
Ausgaben Mitarbeiter	2x monatlich á 2 Stunden	25€	1200€

Tab. 4: Kosten Kunden Mitarbeiter

Beschreibung	Kosten
Monatliche Kosten Mitarbeiter	1200€
Gesamtkosten Auszubildender	840€
Differenz	360€

Tab. 5: Differenzrechnung

Als Vergleich kann man schließen, dass die Individualsoftware sich ab dem 8. Monat rentiert und der Mitarbeiter, der die Daten eingepflegt hat ist nun frei für andere Tätigkeiten.

6 Erstellen eines WordPress Plugins

Beginn eines jeden WordPress Plugins ist erstmal das „Inhaltsverzeichnis“ des Plugins.

Dort wird der Name, die URI, Beschreibung, Version, *Author* und *Author URI* angegeben. Diese Datei sollte am besten auch den Namen des Plugins tragen, damit das System alles reibungslos erkennt. Oft wird diese Datei aber auch einfach nur „Index“ genannt.

Folgend nun ein Standard der schon fast zu einem Klassiker eines jeden WordPress Plugins wurde -> ABSPATH or die.

Als nächstes kommen die einzelnen PHP-Dateien, alle werden mit einem *include_once* in der Index-Datei eingebunden. Zum Abschluss der Index-Datei noch ein paar Variablen, die für die CSV-Dateien benötigt werden und über die Index-Datei dann globalisiert werden im Plugin. Die Funktionen wurden alle in eine separate Datei gepackt, um mehr Sicherheit gewährleisten zu können.

```php
$fmt = numfmt_create( locale: 'de_DE', style: NumberFormatter::DECIMAL );
$cs = ';';          // Zellen bzw. Datenfeld-Trenner
$ts = "\"";         // Text bzw. Datensatz-Trenner
$esc = "\\";
//$eol = "\r\n";    // PHP 8.1
```

Grafik 1: Trennzeichen CSV-Datei

7 Herausfiltern der notwendigen Daten

Hier hat der Autor erstmal rausgesucht, welche Datensätze er für die Erstellung der CSV-Dateien benötigt und des Weiteren auch, welche ich bearbeiten bzw. formatieren muss.

Folgend nun ein kleiner Überblick über die Datensätze, die angepasst wurden:

E-Mail-Adressen	Formatierung für Sonderzeichen
Vor- & Nachname	Formatierung für Sonderzeichen
Adressdaten	Formatierung für Sonderzeichen
Bezahlmethode & Währung	Formatierung für Erkennung der Währung
Zeiten	Formatierung in deutsche Uhrzeit

Tab 6: Zu bearbeitende Daten

Die Felder wurden mit den Funktionen *„mb_detect_encoding“* und *„iconv*“ formatiert, damit Sonderzeichen wie zum Beispiel „äöüß“ erkannt werden.

Des Weiteren hat der Autor aber auch noch die Felder wie Registrierungsdatum, Letzte Aktivität und Letzte Neuerung. Diese „Datumsangaben“ werden mit dem Befehl „date“ in eine deutsche Uhrzeit formatiert, nachdem sie zu einem *„timestamp*“ formatiert wurden.

8 Die Funktionen im Überblick

8.1 Adminmenü

Für das Adminmenü hat der Autor erstmal ein *„add_action*“ benutzt mit dem „Hook-Name (Aktion)“ -> *„admin_menu*“. Folgend hat er seine Funktion *„test_plugin_setup_menu*“ um einen weiteren Menüpunkt und ebenfalls auch Unterseiten zu definieren. *(vgl. Quellen „Callback hinzufügen“)*

Die Funktionen die hier genutzt werden sind *„add_menu_page“* und *„add_submenu_page“.*

Die Strings *page_title*(Hauptseite) und *parent_slug*(Unterseite) sind namentlich an den Praktikumsbetrieb gehalten, restliche Namen/Titel sind vom Namen her eher an die Funktion gehalten.

Die einzelnen Callback Funktionen sind die erstellten Funktionen von ihm, die er ebenfalls dann mit Ausgabefunktionen wie *„echo*“ etc. erweitert hat um eine Ausgabe zu generieren, damit der Kunde auch eine Mitteilung bekommt, was dort passiert.

Diese Funktion beinhaltet 5 Seiten. Die Hauptseite bzw. „Titelseite“, die geschriebenen Text für das Plugin enthält.

Die 4 Unterseiten teilen sich die Funktionen *„customercsv“, „ordercsv“, „directory_creation“* & die API.

8.2 Directory

Als erstes musste der Autor überprüfen, in welchem Directory er sich befindet.

Dafür gibt es die Funktion __*DIR*__ in PHP. Diese hat der Autor sich ausgeben lassen mit einem „echo“ und folgend konnte er dann sehen, in welchem Ordner er sich befindet und ist dann Schritt für Schritt zu der Ordnerstruktur wo hinwollte!

Nachdem der Pfad für die Ordnerstruktur soweit fertig gestellt wurde, geht es nun daran, folgt nun die Erstellung der Ordner.

Dafür wurde u.a. die Funktion *„file_exists“* genutzt, um zu prüfen, ob der Ordner bereits existiert.

```php
if ( ! file_exists( filename: __DIR__ . '/../../../uploads/kisys_export/customer' ) ) {
    mkdir( directory: __DIR__ . '/../../../uploads/kisys_export/customer' );
    echo 'Verzeichnis customer erfolgreich erstellt';
    echo nl2br( string: PHP_EOL );
} else {
    echo 'Verzeichnis customer bereits vorhanden';
    echo nl2br( string: PHP_EOL );
}
```

Grafik 2: Erstellung von Verzeichnis

Nach erfolgreicher (Kein Ordner vorhanden) „If“ Abfrage wird dann mit „mkdir“ zuerst der Oberordner erstellt mit dem Namen des Plugins.

Des Weiteren sendet sowohl die positive als auch die negative „If“ Abfrage noch ein „echo“ Befehl, der dann ausgibt „Ordner erstellt“ oder „Ordner existiert bereits“.

Nach dem „Oberordner“ folgen natürlich noch weitere Ordner. Nämlich für die Order_CSV & Customer_CSV.

Des Weiteren wird an diese Datei auch noch ein register_activation_hook drangehangen, der direkt die directory.php als File anspricht und als Callback Funktion die Funktion „directory_creation“.

8.3 Erklärung CSV

Was bedeutet CSV?

Die Abkürzung CSV bedeutet comma-separated values oder character-separated values.

Eine CSV-Datei ist immer eine Tabelle. Zur Folge hat sie einen Header (Überschrift) und die jeweiligen Content Spalten.

Aufgrund von leichtem Import & Export einer CSV-Datei für zum Beispiel Microsoft Excel hat der Autor sich hierfür entschieden, da der Auftraggeber einen Ausdruck dieser Daten (u.a. für WW(Warenwirtschaft)) benötigt.

Was man vorbereiten sollte/könnte:

Der Autor hat sich zuerst mit der Reihenfolge beschäftigt, in der die Daten dort eingetragen werden.

In der „Index-Datei“ des Plugins wurden die Trennzeichen als Variablen deklariert, in der „*ordercsv.php*“ bzw. „*customer.csv*“ wurden sie dann nochmal globalisiert.

Für den Header hat der Autor einen Array erstellt, der die einzelnen Titel/Überschriften in sich hat.

Danach hat er für jegliche Angaben, die für eine CSV-Datei benötigt werden eine entsprechende Variable angelegt. Zum Beispiel ist der Pfad, wo die CSV-Datei gespeichert werden soll eine Variable. Ebenso aber auch der Name der Datei, wie zum Beispiel „*order.csv*“. Des Weiteren hat er die Trennzeichen in einer separaten Datei deklariert, damit man sie danach nur noch in die entsprechende Datei globalisieren muss zum „benutzen“.

Jetzt wo die Vorbereitungen alle soweit abgeschlossen sind, kann man nun starten mit dem befüllen der CSV-Datei.

Wie man eine CSV-Datei bearbeitet:

Zu Beginn öffnet der Autor erstmal das entsprechende Directory -> folgend wird überprüft, ob man im Verzeichnis ist und der „Modus“ wird ausgewählt, welcher bestimmt, wie die CSV-Datei zu bearbeiten ist. Der Modus „w“, der ausgewählt wurde öffnet die Datei zum Schreiben und setzt den Dateizeiger auf den Anfang der Datei. Wenn die Datei nicht existiert wird diese erstellt.

Gesetz dem Fall, der Ordner kann nicht geöffnet werden oder die Datei kann nicht erstellt werden ist ein entsprechendes „die“ vorhanden mit einer Fehlermeldung.

Als nächstes wird die „Kopfzeile“ in die CSV-Datei eingetragen, danach über eine foreach Schleife dann die Inhalte.

```php
// Headline/Line -> CSV
fputcsv( $fp, $labelForOrder, $cs, $ts, $esc );
// Data -> CSV
foreach ( $arrOrder as $arrOrder ) {
    if ( $arrOrder['Status'] === 'In Wartestellung' ) {
        fputcsv( $fp, $arrOrder, $cs, $ts, $esc );
    }
}
```

Grafik 3: Kopfzeile und Inhalte in CSV schreiben

Wenn soweit alles funktioniert hat, gibt das Skript am Ende noch eine Bestätigungsmeldung raus.

8.4 Order_CSV

Folgende Datensätze werden benötigt:

Id	Bestell ID
Customer id	ID des Kunden
Status	Status der Bestellung
Currency	Währung
total	Gesamtpreis der Bestellung
Payment method title	Wie bezahlt wird (z.B. Kreditkarte, Rechnung)
Date created	Wann die Bestellung eingegangen ist
Name	Produktbezeichnung
Quantity	Anzahl der Produkte
subtotal	Preis eines Produkts welches mehrfach bestellt wird
sku	Produkt ID

Tab. 7: Benötigte Inhalte der „*order.csv*“

Folgend sehen Sie eine selbsterstellte Grafik, die den Zyklus einer Bestellung dargestellt mit den entsprechenden Schritten, die durchlaufen zu sind.

Grafik 4: Zyklus einer Bestellung in der Funktion

Gestartet wird mit der Sortierung des Headers. Nachdem das erledigt ist und die Variablen als Vorbereitung schon erstellt sind, folgt das Erstellen der CSV-Datei.

Es wird der Pfad geöffnet, wo die Datei gespeichert werden soll. Danach wird noch eine Gegenprüfung gemacht um Fehler zu vermeiden. Nun wird der Header inklusive globalisierter Trennzeichen in die CSV-Datei geschrieben mit dem Befehl „fputcsv“.

Da dieser Vorgang soweit erfolgreich war, beginnen wir nun, den inhaltlichen Content in die Datei zu schreiben. Dies geschieht mit einer „foreach“ Schleife.

Dadurch, dass der Autor bereits beendete Bestellungen nicht anzeigen lassen möchte, deklariert er in der „foreach“ Schleife vor dem schreiben in die CSV-Datei noch eine „If“ Abfrage, wo die

Anforderung ist „Bestellstatus = In Wartestellung“. Nun werden nur entsprechende Bestellungen in die Datei eingetragen und die restlichen Bestellungen (wie zum Beispiel stornierte) werden ignoriert.

Jetzt folgt die Bestätigung, dass die Bestellung in die CSV-Datei geschrieben wurde, die Schließung der Datei und die Schließung des Verzeichnisses (Directory).

Als krönenden Abschluss folgt nun der WordPress eigene PHP-Befehl „wp_reset_postdata“, um die $post global für den aktuellen Beitrag wieder her zu stellen.

8.5 Customer_CSV

Zu Beginn eine Tabelle, wo man sieht welche Parameter für die CSV-Datei benutzt und ggf. bearbeitet bzw. formatiert wurden.

id	wird nicht bearbeitet
Display_name	wird nicht bearbeitet
User_registered	wird in deutsche Zeit umgewandelt
User_email	wird formatiert
Last_update	wird in deutsche Zeit umgewandelt
Wc_last_active	wird in deutsche Zeit umgewandelt
Billing_first_name	wird formatiert
Billing_last_name	wird formatiert
Billing_address_1	wird formatiert
Billing_postcode	wird formatiert
Billing_city	wird formatiert
Billing_country	wird formatiert
Billing_phone	wird nicht bearbeitet
Billing_email	wird formatiert
Shipping_first_name	wird formatiert
Shipping_last_name	wird formatiert
Shipping_address_1	wird formatiert
Shipping_postcode	wird formatiert
Shipping_city	wird formatiert
Shipping_country	wird formatiert
Shipping_phone	wird nicht bearbeitet

Tab 8: Inhalte der „*customer.csv*“

Wie auch bei der Order-CSV hat der Autor hier erstmal damit begonnen, sich die Variablen zu erstellen, die Trennzeichen zu globalisieren und den Header in die richtige Reihenfolge zu bringen. Als dies erledigt war hat er die User ebenfalls versucht rauszuschreiben, sobald sie existieren oder sich im Profil etwas ändert.

Über ein Array „args“ hat er dann die Filterung übernommen, dass nur „Customer“, also Kunden herausschreibt. Administratoren werden hier nicht berücksichtigt.

Die Schwierigkeit hat sich dann herausgestellt, dass der Kunde möchte, dass man sich nicht automatisch registrieren muss um bestellen zu können. Dementsprechend musste er dann erstmal schauen, wie man Userdaten noch bekommt, außer über die Funktion „get_users“.

```
iconv(mb_detect_encoding($user->data->display_name), to_encoding: 'Windows-1252//ORIGINAL',$user->data->display_name),
```

Grafik 5: Konvertierungsvorgang des „Anzeigename“

Er hat dann herausgefunden, dass man sich in WordPress bei den Bestellungen über die „get_user_meta“ die Daten von nicht registrierten Benutzern holen kann. Dadurch, dass er aber ohne Schlüssel keine Daten ausgegeben wird, musste er über die Funktion „array_key_exists“ auf die Daten zugreifen. Des Weiteren um Schwachstellen zu vermeiden oder fehlerhafte Ausgaben, wurden einige Datensätze entsprechend formatiert.

```
$meta = get_user_meta($user->ID);
if( array_key_exists( key: 'last_update', $meta ) ){
    $row['Letzte_Neuerung'] = date( format: 'd.m.Y H:i:s', $meta['last_update'][0] );
} else {
    $row['Letzte_Neuerung'] = '';
}
```

Grafik 6: Formatierung der Zeitangaben

Als er dann alle Datensätze in entsprechender Ausgabe bekommen hat, wurden die Daten in die CSV-Datei eingepflegt. Angefangen mit dem Header, um die richtige Reihenfolge der Daten zu gewährleisten, folgend von den Trennzeichen und dann die einzelnen User/Käufer-Daten.

Nachdem dieser Vorgang beendet war, hat er sich daran begeben, die Erstellung der CSV-Datei so zu manipulieren, dass man jederzeit sieht, ob sich was an den User-Daten ändert (zum Beispiel Firmenumzug). Daraufhin hat er mit *scandir* erstmal den Ordner überprüft um zu sehen, ob sich dort versteckte Dateien aufhalten. Der Autor hat dort eine „.“ und eine „..“ Datei gefunden.

Dieses Ergebnis hat er mit *array_diff, natsort* und *array_slice* bearbeitet, damit er nur noch meine erstellten Dateien angezeigt bekomme.

Ebenfalls hat er den Namen der CSV-Datei aufgeteilt, um mehrere Dateien mit der Erweiterung eines Counters zu erstellen, wo er die Möglichkeit hat die Dateien mit 1, 2, 3 etc. am Ende des Namens zu kennzeichnen.

User/Käufer, die zum Beispiel keine separate Lieferadresse haben, sondern Liefer- und Rechnungsadresse zusammen ist, die Felder werden leer gelassen.

8.6 Cronjobs

Ein Cron ist ein automatisierter Prozess, der zeitbasiert wiederkehrende Aufgaben ausführt. Meistens unter Linux-Systemen zu finden, aber auch im Web ein nützliches Tool, wenn es um die Bearbeitung von Dateien geht. *(vgl. Quellen Erklärung Cronjob)*

Um die Arbeit zu erleichtern, wurde das Plugin „WP Crontrol“ auf das System aufgespielt. (*vgl. Quellen WP-Crontrol)*

Im Admin-Menü unter „Werkzeuge“ ->Cron Events sieht man die akut laufenden Cron Events.

Unter dem Punkt „Einstellungen“ -> Cron Schedules kann man ein Cron Event erstellen.

Wenn man unter Cron Events dann den neu erstellten Cron editiert -> Hier kann man jetzt einstellen, wann der Cron starten soll.

Die Funktion, an die der Cron dran geheftet werden soll, findet man in der „*cronjobs.php*“.

Dort hat man mit je einem *„add_action“* die Cron Events für das System aktiviert und erstellt für jeden Cron nun eine eigene Funktion, die dann wiederrum eine weitere Funktion aktiviert bzw. reaktiviert.

Dailycron: *24 Stunden Cron*

Crontest: *5 Minuten Cron*

```
add_action( 'crontest', 'my_cron_schedules2' );
function my_cron_schedules2() {//Alle 5 Minuten
status_changing();
}
```

Grafik 7: Codeschnipsel Cronjob

8.7 Status_Changing

Diese Funktion ist dafür da, den Status der Bestellung nach 24 Stunden zu ändern.

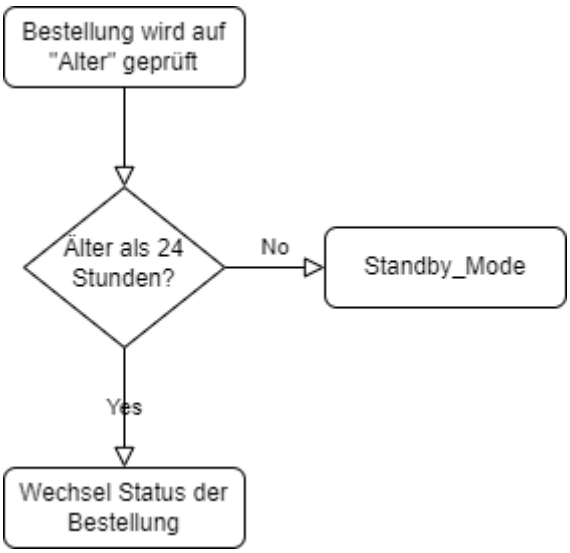
Angefangen mit einem *add_action* wird an die Datei noch ein Cronjob eingebunden, der alle 5 Minuten den Code „triggert“. Als Callback natürlich die Funktion *„status_changing“*.

In der Funktion habe ich zuerst eine Variable *testdate* definiert, die mit der Funktion *„time“* (Unix-Zeitstempel), wo das Plugin -1 Tag rechnet (*time*()-60*60*24).

Folgend hat der Autor einen Array *„args“* erstellt, in der zum einen den Bestell-Status filtere auf „on-hold“ (In Wartestellung) und ebenfalls wurde dort noch einen Vergleich eingestellt, der den Zeitpunkt der Erstellung(*date_created*) < Variable *testdate* vergleicht.

Danach wurde in einer *„foreach* Schleife“, die über die Bestellungen läuft *(wc_get_orders*) eine *„If“* Abfrage, die sich erst den Bestell-Status holt, ihn dann überarbeitet (*updated*) und danach speichert.

Die hier folgende Grafik dient zur Veranschaulichung der Funktion.



Grafik 8: Grafische Darstellung der Funktion

8.8 Rest_API

8.8.1 Erklärung Rest API

REST = Representational State Transfer

REST vereinfacht im Gegensatz zu anderen Verfahren die Datenübertragung und konzentriert sich aufs wesentliche. Im vorliegenden Fall reicht dies aus. Des Weiteren liegen alle notwendigen Technologien im Internet bereits zugrunde.

Initial ist es notwendig, zur Verbindung einen API-Schlüssel anzulegen. Dazu muss man im Admin-Menü der WordPress Seite unter WooCommerce->Einstellungen->Erweiterungen->Rest-API einen Schlüssel hinzufügen, dann fragt das System nach einer Beschreibung, für welchen Account der Schlüssel sein soll und welche Rechte darauf sein sollen (Lesen, Schreiben, Beides). Nachdem beides ausgewählt wurde klickt man jetzt auf API-Schlüssel erstellen. Dieser Vorgang kann je nach System 1-2 Minuten in Anspruch nehmen.

Die Datenpflege erfolgt in der Warenwirtschaft. Der Webshop ist mandantenfähig und verwaltet somit mehrere Kunden. (beide Web-Shops)

Folgend werden die Daten per API von der Warenwirtschaft an den Web-Shop gesendet, der als Master-Shop fungiert. Das erstellte Plugin befindet sich im Master-Shop.

Alle Produkte die im 2. Web-Shop angelegt werden, sollen via. API über eine Produkt-Kategorie ausgewiesen werden. Gesetz dem Fall, ein Produkt ändert sich (Preis, Name o.Ä.) wird die Korrektur übertragen.

Ebenfalls den Bestandsstatus fängt der Autor ab, um zwischen den jeweiligen Shops keine Fehlbestände zu erhalten. Des Weiteren werden Bestellungen direkt mit der Warenwirtschaft ausgetauscht und verarbeitet. Somit ist die Bestandsanpassung seitens des Master-Shops immer gegeben.

9 Erweiterungsmöglichkeiten

Eine sehr schnelle & leichte Erweiterung wäre, weitere bzw. andere Produkte / Produktgruppen von dem Partner-Shop zu beziehen.

Ebenfalls können auch weitere Datensätze herausgeschrieben werden oder für weitere Zwecke andere CSV-Dateien erstellt werden. Des Weiteren kann man auch noch einen oder mehrere Buttons hinzufügen, um die ganzen automatisierten Vorgänge auch separat manuell machen zu können.

Mit eine der besten und schnellsten Erweiterungsmöglichkeiten ist es, wenn der Kunde noch mehr „Partnerschaften“ mit anderen Online-Shops gründet. Dies könnte dann mit nahezu selbem Code mit sehr geringem Aufwand erweitert werden.

Was auch schon im Gespräch war, bisher aber noch nicht geschehen ist: Den Shop „Mandantenfähig“ (*vgl. Quellen Mandantenfähigkeit*) zu machen. Das heißt, dass der Kunde sich auf die Schnittstelle eines anderen Systems oder direkt auf das ganze System bezieht.

10 Fazit

Sein Fazit dazu ist, dass man nichts unterschätzen sollte, auch wenn es zumindest theoretisch selten genutzt wird. PHP ist eine mächtige Programmiersprache und er ist froh, einen weiteren Einblick in die Funktionen werfen zu dürfen. Er konnte viel dazu lernen und konnte Einblicke in jegliche Bereiche in der Web-Entwicklung bekommen.

Als Lesson learned kann man sagen, dass im Bereich der Web-Entwicklung mit Plattformen wie WordPress eine Menge Eigenfunktionen und Abgrenzungen gibt, in die sich der Autor erstmal einarbeiten musste. Ebenfalls hat er Interesse an der API entwickelt, welches er weiter ausbauen möchte.

Rückblickend sieht der Autor auch keine großen Optimierungen, sondern würde es gerne erweitern und somit das Risiko verringern, zu viele Plugins zu nutzen und weniger Angriffsfläche zu bieten.

Des Weiteren findet er es interessant, welche Möglichkeiten sich ergeben bei doch relativ einfachen Befehle

11 Anhang

11.1 Quellenangaben

Erklärung Cronjob: https://de.wikipedia.org/wiki/Cron (Stand: 05.09.2022)

Mandantenfähigkeit: https://de.wikipedia.org/wiki/Mandantenfähigkeit (Stand: 11.09.2022)

Marktanteile und Nutzungsstatistiken: https://de.statista.com/statistik/daten/studie/320670/umfrage/marktanteile-der-content-management-systeme-cms-weltweit/#:~:text=Marktanteile%20der%20Content%2DManagement%2DSysteme,CMS)%20weltweit%20im%20September%202022&text=Bis%20zum%2019.,rund%2064%2C2%20Prozent%20an. (Stand: 03.10.2022) & https://w3techs.com/technologies/details/pl-php (Stand: 03.10.2022)

WP-Crontrol: https://de.wordpress.org/plugins/wp-crontrol/ (Stand: 02.09.2022)

WooCommerce Bestelldetails:

Explizite Plugin Funktionen: https://developer.wordpress.org/reference/functions/register_activation_hook/ (Stand: 01.09.2022)

WordPress Entwicklerfunktionen: https://stackoverflow.com/questions/39401393/how-to-get-woocommerce-order-details (Stand 02.09.2022) & https://github.com/woocommerce/woocommerce/wiki/wc_get_orders-and-WC_Order_Query (Stand 02.09.2022)

WordPress Manual:

Plugin bei Aktivierung starten: https://developer.wordpress.org/reference/functions/register_activation_hook/ (Stand: 05.09.2022)

Callback hinzufügen: https://developer.wordpress.org/reference/functions/add_action/ (Stand: 05.09.2022)

User-Metadaten abrufen: https://developer.wordpress.org/reference/functions/get_user_meta/ (Stand: 06.09.2022)

Resetten der Post-Variable: https://developer.wordpress.org/reference/functions/wp_reset_postdata/ (Stand: 08.09.2022)

Rest API Funktionsbibliothek: https://woocommerce.github.io/woocommerce-rest-api-docs/#libraries-and-tools (Stand: 05.09.2022)

PHP-Manual:

Datei öffnen und bearbeiten: https://www.php.net/manual/de/function.fopen.php (Stand 02.09.2022) & https://www.php.net/manual/de/function.fputcsv.php (Stand: 02.09.2022)

Versionsupdate: https://www.php.net/supported-versions.php (Stand: 06.09.2022)

Erstellung von Grafiken: https://app.diagrams.net/ (Stand: 01.09.2022)

Datei in einen String lesen: https://www.php.net/manual/de/function.file-get-contents.php (Stand: 11.09.2022)

Konvertierung der Zeichenkette: https://www.php.net/manual/de/function.iconv (Stand: 11.09.2022)

Überprüfung der Zeichenkette: https://www.php.net/manual/de/function.mb-detect-encoding.php (Stand: 11.09.2022)

Dekodierung von JSON: https://www.php.net/manual/de/function.json-decode.php (Stand: 07.09.2022)

JSON Online Validerer: https://jsonformatter.org/ (Stand: 07.09.2022)

11.2 PHP-Funktionen

Hier finden Sie einen Überblick über die PHP eigenen Funktionen die dafür genutzt wurden:

Funktion	Beschreibung
echo	Gibt einen oder mehrere Strings aus
print	Ausgabe eines Strings
print_r	Gibt Variablen lesbar aus
if	Eine „Wenn“ Abfrage
else	Antwort, falls „Wenn“ Abfrage nicht stimmt
foreach	Funktion, um Arrays zu iterieren
exit	Beendet den Vorgang
die	Beendet den Vorgang
date	Formatiert einen Unix-Zeitstempel
time	Gibt den aktuellen Unix-Zeitstempel
strtotime	Konvertiert Datums- und Zeitangabe zu Unix-Zeitstempel
fopen	Öffnet eine Datei
fputcsv	Befüllt eine Datei
fclose	Schließt eine Datei
opendir	Öffnet einen Ordner
scandir	Scannt einen Ordner
closedir	Schließt einen Ordner
mkdir	Erstellt einen Ordner
file_exists	Prüft, ob eine Datei existiert
strlen	Ermittelt die inhaltliche Länge eines Strings
substr	Liefert einen Teil eines Strings
count	Zählt
array_slice	Extrahiert einen Ausschnitt eines Arrays
array_diff	Vergleicht 2 Arrays
natsort	Sortiert einen Array in „natürliche Reihenfolge“
nl2br	Sorgt für einen Zeilenumbruch
pathinfo	Liefert Informationen über einen Dateipfad
array_key_exists	Prüft, ob ein entsprechender Schlüssel in Array existiert
mb_detect_encoding	Zeichenkodierung erkennen
iconv	Konvertiert eine Zeichenkette
__DIR__	Zeigt das Directory, in dem man sich befindet
json_decode	Dekodiert eine JSON-Zeichenkette
file_get_contents	Liest die Datei in einen String
include_once	Bindet eine Datei ein und führt sie aus
numfmt_format_currency	Formatiert einen Währungswert

Tab. 9: Benutzte PHP-Funktionen

11.3 WordPress Funktionen

Hier sehen Sie einen Überblick über die WordPress Funktionen die benötigt wurden:

Funktion	Beschreibung
add_menu_page()	Im Adminmenü einen Menüpunkt erstellen
add_submenu_page()	Im Adminmenü eine Unterseite erstellen
add_action()	Fügt einem Aktions-Hook eine Rückruffunktion hinzu
get_option()	Ruft einen Optionswert entsprechend dem Namen ab
update_option()	Aktualisiert den Wert einer Option, die bereits hinzugefügt wurde
wc_get_orders()	Bestellungen aus System bekommen
get_items()	Bestellte Produkte bekommen
get_id()	WordPress ID bekommen
get_sku()	Produkt ID bekommen
get_customer_id()	Kundennummer bekommen
get_currency()	Währung bekommen
get_quantity()	Artikel Menge bekommen
get_total()	Gesamtsumme bekommen
get_subtotal()	Teilsumme bekommen
get_status()	Status bekommen
update_status()	Status „updaten“ (aktualisieren)
save()	Speichern
get_payment_title()	Bezahlmethode bekommen
get_date_created()	Erstellungsdatum bekommen
get_users()	User aus System bekommen
get_user_meta()	User aus Bestellung bekommen
wp_reset_postdata()	Nach Durchlaufen einer separaten Abfrage stellt diese Funktion die $post global für den Beitrag in der Hauptabfrage wieder her
register_activation_hook()	Legen Sie den Aktivierungshaken für ein Plugin fest

Tab. 10: Benutzte WordPress Funktionen

11.4 Geschichte von PHP

Version 1.0 erschien am 8. Juni 1995.

Zu Beginn hieß es nicht PHP, sondern PHP FI (Personal Home Page – Formas Interpreter).

Der Gründer von PHP FI heißt Rasmus Lerdorf. Im Jahre 1997 wurde Rasmus L. von zwei Studenten mit den Namen Andi Gutmans und Zeev Suraski kontaktiert, um mehrere Themen anzusprechen und zusammen entwickelten Sie auf Basis von PHP/FI 2.0 das heutige PHP mit der Version 3.0.

Jede Version von PHP hat einen 2 Jahre aktiven Support und ein weiteres Jahr Sicherheitsunterstützung.

Die aktuell letzte Version von PHP ist 8.1.7 und erschien am 9. Juni 2022.

Bekannte Websites, die PHP verwenden sind Facebook, Wikipedia & WhatsApp.

11.5 Geschichte von WordPress

WordPress ist ein Content Management System, kurz CMS, welches hilft, Inhalte zu bearbeiten ohne typische HTML Editor Software nutzen zu müssen. Des Weiteren ist es Open Source Software und lizensiert unter GPLv2. Das heißt, dass es frei zu benutzen und zu verändern ist.

Die Gründer heißen Matt Mullenweg, Mike Little. Die Freundin von Matt (Christine Tremoulet) ist die Namensgeberin.

Am 1. April 2003 erschufen sie einen neuen Zweig des alten Systems „b2“. Die erste Version (0.7) kam am 27. Mai 2003. Die offizielle Version 1.0 mit dem Namen „Miles Davis“ erschien am 3. Januar 2004. Mit den Versionsnamen haben die Entwickler ihre Bewunderung und Hingabe zur Jazz-Musik geteilt.

Des Weiteren existiert in jeder Veröffentlichung von WordPress ein Plugin namens „Hello Dolly“. Dies ist eine Hommage an Louis Armstrong.

Folgend nun ein Zitat von Matt Mullenweg:

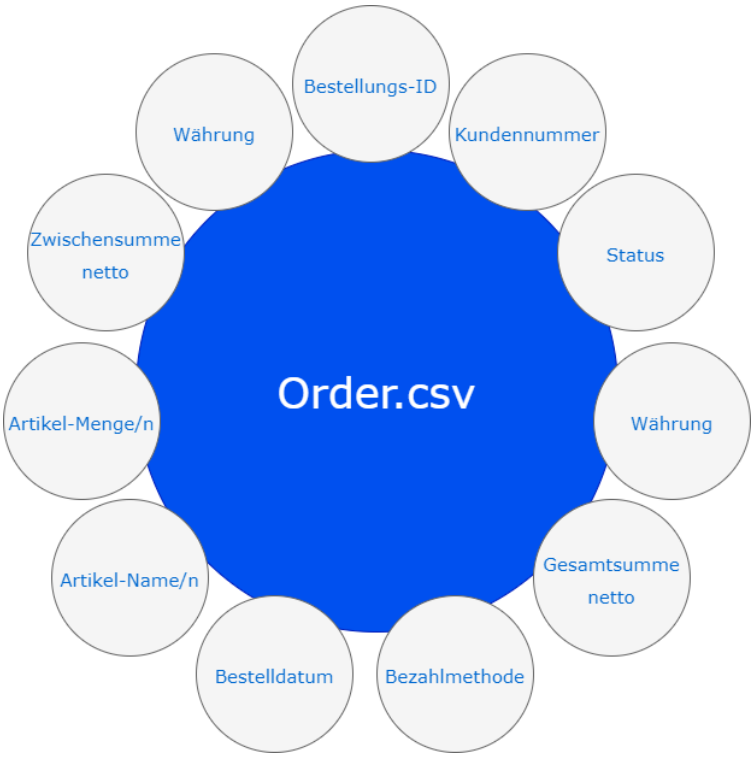
„Das Hello Dolly Plugin symbolisiert die Hoffnung und den Enthusiasmus einer ganzen Generation, zusammengefasst in zwei Worten, die Louis Armstrong am berühmtesten gesungen hat.“

Die Version 6.0 mit dem Namen „Arturo O´Farrill“ am 24. Mai 2022 ist die aktuell letzte.

Mit über 60% Marktanteil der Content Management Systeme ist WordPress ein Oligopol. Die genaue Statistik dazu befindet sich im Anhang Seite 17.

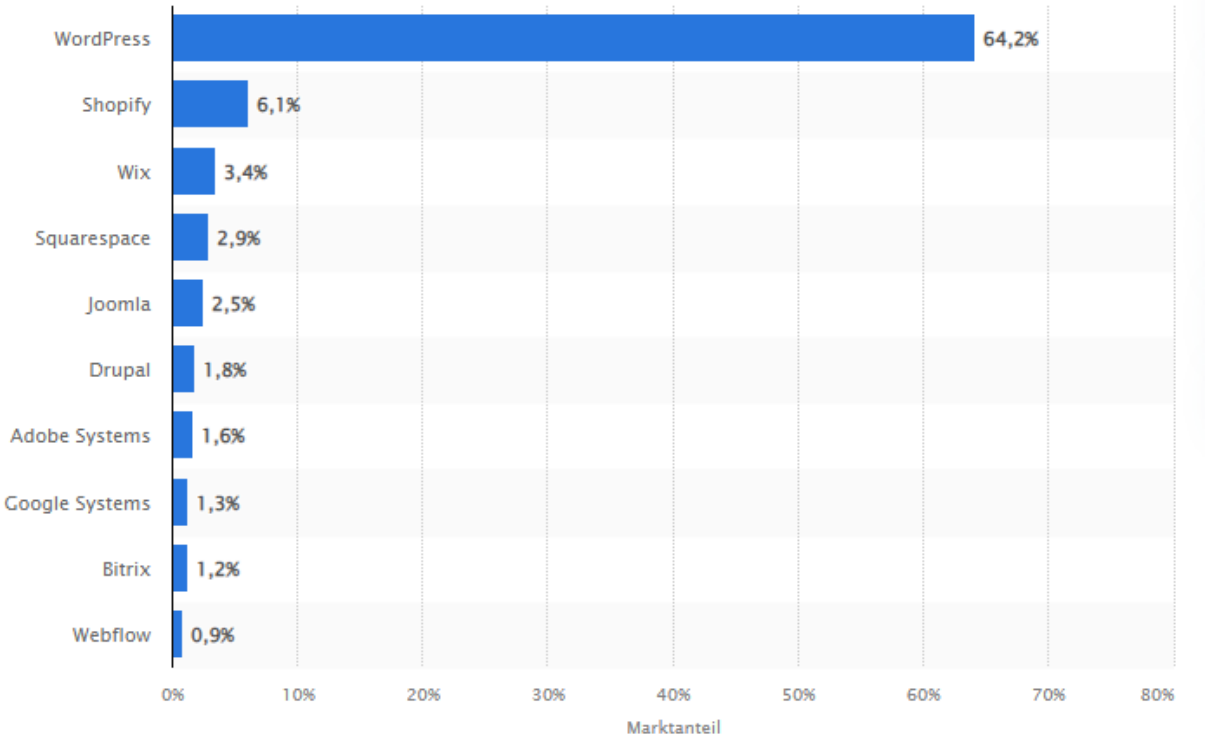
11.6 Weitere Grafiken

Die Datei order.csv mit ihrem Inhalt grafisch dargestellt.



Grafik 9: Optische Darstellung „order.csv“

Hier eine genaue Statistik der Top 10 Content Management Systeme aus dem Jahr 2022.

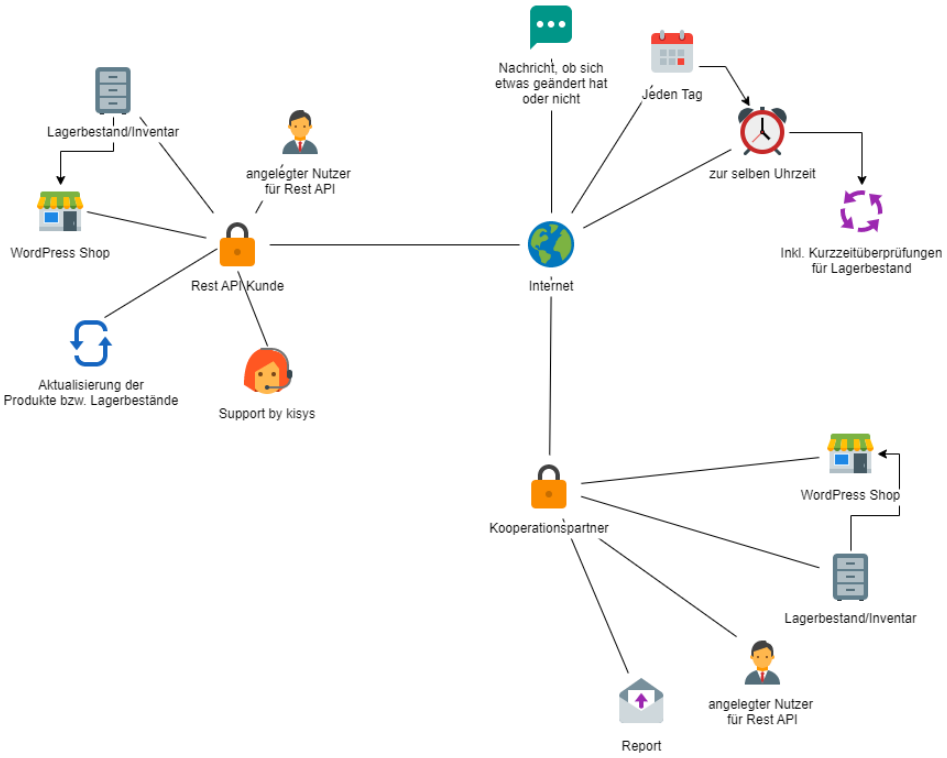


Grafik 10: Bewertung der CMS-Systeme 2022

Hier sieht man eine kleine grafische Anleitung, wie man sich mit der Rest API verbindet.

Grafik 11: Kurzerklärung Rest API

Hier eine grafische Darstellung, was genau im System passiert für die API Prüfung.



Grafik 12: Schaubild der Funktion