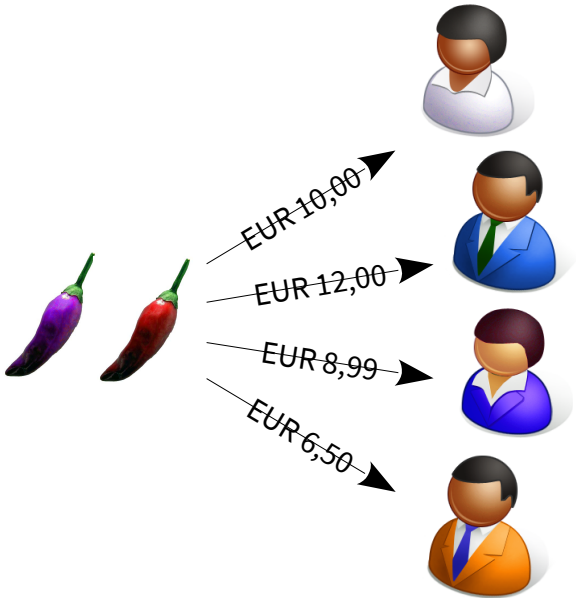
PEPPERSHOP

PepperShop Preisfindung Modul

Anleitung



Datum
31. Juli 2017

Version
2.1

PEPPERSHOP

Inhaltsverzeichnis

ACHTUNG: Dieses Modul benötigt Customizing.
Es werden PHP-Kenntnisse vorausgesetzt.

PepperShop wird von Glarotech entwickelt und vertrieben. Seit 1998 ist das innovative Unternehmen im Internet tätig und auf E-Commerce spezialisiert. Sie als Kunde profitieren vom direkten Draht zu den Herstellern der Produkte.

glarotech

Glarotech GmbH
Toggenburgerstrasse 156
CH-9500 Wil

info@glarotech.ch
Tel. +41 (0)71 923 08 58
www.glarotech.ch

1. Einleitung

Mit diesem Modul setzte man Preisfindungsregeln im PepperShop um. Im Gegensatz zu den Rabattierungsregeln, die erst im Warenkorb angezeigt werden, läuft die Preisfindung schon im Artikelkatalog und zeigt somit jedem angemeldeten Kunden sofort seinen eigenen Preis.

Preisfindungsregeln sind sehr flexibel einsetzbar und z.B. auch per Geo-IP[1] Modul kombinierbar. Sie gehen noch einiges weiter: Man kann Artikel eine aktive Aktion vergeben oder sie gar als nicht käuflich markieren.

Info 1: Zusätzlich zu diesem Preisfindungsmodul gibt es für die Enterprise Shopversionen auch noch das umfangreiche Modul "Debitorenpreise und Zeilenrabatte"[2]. Dies setzt eine sehr komplexe, Datenbank basierte Preisfindung um, welche per Import komplette Preisfindungsregelungen aus ERP-Systemen übernehmen und aktualisieren kann.

Info 2: Zusätzlich kann man mit dem Enterprise Modul "Best Price Findung"[2] ein System im PepperShop in Betrieb nehmen, welches den jeweils besten Preis für den Kunden ausfindig macht (Aktion / Mengenrabatte / Gutscheinkampagnen / …).

2. Funktionsweise des Moduls

2.1 Art der Umsetzung / PepperShop Internals

Die Preisfindung ist ziemlich invasiv und sehr tief im Shopsystem verankert. Die Umsetzung wurde so gestaltet, dass man trotzdem die gesamte Preisfindungsregelung an nur einer Stelle im Shop regelt und somit die Artikel vielfältig beeinflussen kann. Grundsätzlich erhält man jeden Artikel, bevor er angezeigt wird in einer Methode präsentiert und kann hier nach belieben Anpassungen vornehmen (Aktionen beeinflussen, Promoflag setzen, Beschreibungen ändern, Käuflichkeit anpassen, oder auch einfach nur den Preis überschreiben).

Die Anwendung des Moduls kann extrem einfach sein, aber auch komplex, da gegebenenfalls sehr Performance sensitiv / mit Caching gearbeitet werden muss, oder man sich über die internen Berechnungsabläufe der einzelnen Subsysteme Gedanken machen muss.

Als Basis dient die Funktion getArtikel(...). Wenn diese Funktion verwendet wird, kann man davon ausgehen, dass die Preisfindung durchlaufen worden ist. Ebenso bei einigen Aggregatsfunktionen für die Artikellistendarstellung im Katalog. Da der Warenkorb ebenfalls mit diesen API-Funktionen operiert ist der gesamte Bestellprozess ebenfalls abgedeckt.

Wo man extrem vorsichtig sein muss, sind selbst erstellte, direkte Datenbankzugriffe auf die `artikel` Tabelle der Shop-Datenbank - hier läuft *KEINE PREISFINDUNG!*

2.2 Überblick einer Umsetzung

Die Definition der Preisfindungsregeln (Logik)

Datei: `{shopdir}/shop/preisfindung.def.php`

In dieser PHP-Datei wird die PHP-Klasse `preisfindung` und darin die Methode `set_artikel-preis` definiert. In dieser Methode wird die Preisfindungs-Logik ausprogrammiert.

Da diese Methode von sehr unterschiedlichen Orten im Shop angesteuert wird, steht der jeweilige Artikel in völlig verschiedenen Containern zur Verfügung: `Artikel` Objekt `Artikel_info`

1 Geo-IP Modul : http://www.peppershop.com/webshop/geoip-zmfunk2.html
2 Preisfindungsmodule : http://www.peppershop.com/webshop/preisfindung-spa.html

Objekt, `dyn_artikelinfo` Objekt, assoziativer Array mit DB-Felddaten der `artikel` Tabelle / mit und ohne Standard PepperShop Varianten in div. Encodierungen.

Damit man ohne grossen Konvertierungsaufwand programmieren kann, stellt das Preisfindung Modul innerhalb des `preisfindung` Objekts einige Membervariablen immer zur Verfügung, unabhängig davon, ob die übergebenen Daten alles beinhalten. Dies wird möglichst Performance sensitiv gemacht und sollte für die meisten Anforderungen bereits ausreichen.

Zur Programmierung der Logik wird weiter unten in diesem Dokument genauer eingegangen.

3. Ausprogrammierung der Logik

Wie im Kapitel "Überblick einer Umsetzung" vorhin erwähnt, wird die Logik zur Sichtbarkeitssteuerung in der Datei `{shopdir}/shop/preisfindung.def.php` in den Methoden `set_artikelpreis` ausprogrammiert.

In dieser Methode wird die eigentliche Logik programmiert. Die als Referenz übergebene Artikel Ressource wird innerhalb dieser Methode den Wünschen entsprechend angepasst und wird somit auch für den weiteren Verlauf mutiert weiter verwendet werden.

Als einziges Argument erhält man die Variable `$artikel_obj_arr` Darin ist entweder ein assoz. Array oder ein Artikel spezifisches Objekt (siehe Kapitel 3.1), abhängig, von wo die Preisfindung aufgerufen worden ist. In der Artikelliste ist es z.B. ein Artikel Objekt. Im Warenkorb ein Artikel_info Objekt oder ein dyn_artikelinfo Objekt, gegebenenfalls auch nur ein assoziativer Array mit Artikeldaten direkt aus der artikel Tabelle der Shop-Datenbank.

Folgende Membervariablen sind im preisfindung Objekt immer abgefüllt

```
$this->eingeloggt        true -> der Kunde ist eingeloggt
                         false -> der Kunde ist (noch) nicht eingeloggt
$this->kunde_obj         Kundenobjekt, wenn der Kunde eingeloggt ist, sonst false
$this->kundengruppe_id   Kundengruppen-ID (0 -> standard)
$this->kunden_nr         Kunden-Nr, wenn der Kunde eingeloggt ist und eine Kundennummer
                         hat, sonst Leerstring
$this->kunden_id         Kunden-ID, wenn der Kunde eingeloggt ist, sonst Leerstring
$this->waehrung          gewaehlte Waehrung des Kunden (eur, chf, ..)
$this->default_waehrung  Default-Waehrung des Shops (eur, chf, ..)
```

"Pointer" auf vorhandene Artikeldaten

Folgende Membervariablen sind an dieser Stelle abgefüllt verfügbar und repräsentieren Pointer auf den Artikel, die direkt beschrieben werden können:

```
$this->art_id                Artikel-ID
$this->art_nr                Artikel-Nummer
$this->art_zusatzfeld_1      Artikelzusatzfeld 1
$this->art_zusatzfeld_2      Artikelzusatzfeld 2
$this->art_zusatzfeld_x      Artikelzusatzfeld x
$this->art_preis             Pointer auf Artikelpreis
$this->art_aktionspreis      Pointer auf Artikel-Aktionspreis
$this->art_aktion_von        Pointer auf Artikel-Aktions-Start-Zeitpunkt
$this->art_aktion_bis        Pointer auf Artikel-Aktions-End-Zeitpunkt
$this->art_staffelpreise_arr Pointer auf Artikel-Staffelpreis-Array (leerer Array, wenn
                             keine Staffelpreise vorhanden)
$this->art_variationen_arr   Pointer auf Artikel-Variantenpreise-Array (leerer Array,
                             wenn keine Varianten vorhanden)
$this->art_optionen_arr      Pointer auf Artikel-Optionenpreise-Array (leerer Array,
                             wenn keine Optionen vorhanden)
```

```
$this->art_preisinfo_obj     Pointer auf Artikel-Preisinfo-Objekt
```

Zugriff auf alle Kategoriedaten

Zugriff auf das vollständige Kategorie Objekt ist wie folgt möglich:

```
$this->art_obj               Artikel Objekt
```

3.1 Performance-Hinweis

Die Artikelzusatzfelder sind beim Aufruf der Preisfindung aus den Artikelkatalog Aggregatsfunktionen `getArtikellight` und `getMultiArtikellight` meistens nicht im Artikel-Array enthalten. Es wird deshalb beim Zugriff auf die Artikelzusatzfeld-Membervariablen der Preisfindung das Artikelobjekt aus der Datenbank ausgelesen (schlecht für die Performance).

Zur Performanceoptimierung sollte `getArtikellight` und `getMultiArtikellight` so aufgerufen werden, dass die für die Preisfindung benötigten Artikelzusatzfelder bereits im übergebenen Artikel-Array enthalten sind. Der Artikel muss dann nicht durch die Preisfindung aus der DB ausgelesen werden.

Beim Aufruf der Preisfindung durch die oben erwähnten Funktionen wird das Artikel-Objekt auch aus der Shop-Datenbank ausgelesen, sobald auf die Membervariable `art_obj` zugegriffen wird (Lazy). Der Zugriff sollte also *nur dann* erfolgen, wenn auch eine Preisfindung erfolgt (z.B. nicht auch für ausgeloggte Benutzer mit Standardpreisen).

Um Zusatzfelder oder weitere Artikeldaten an die beiden oben genannten Funktionen durchzureichen, muss man in der Datei `{shopdir}/shop/USER_ARTIKEL_HANDLING_AUFRUF.php` im Bereich `$darstellen == 1` an folgenden Ort gehen und den dort definierten Array mit auszulesenden Feldern erweitern (Info: Umbrüche an anderen Orten):

```
// Falls wir die Artikelliste rendern (Zweistufigkeit: Stufe 1), so muessen wir der auslesenden

// Funktion neben Artikel_ID, Artikel_Nr und Name noch mitteilen, welche zusaetzlichen Felder wir auslesen muessen
$artikeldarstellung_zweistufig_auszulesende_zusatzfelder = array('Beschreibung','Kurzbeschreibung',
                                                                 'Bild_klein','Bild_gross','Staffelpreise',

'Preis','Mindestbestellmenge','Mindestlagermenge',

                                                                 'anzahl_einheit','anzahl_nachkomma',
                                                                 'anzahl_anzeige','anzahl_dropdown','promo',
                                                                 'Aktionspreis','Aktion_von','Aktion_bis',
                                                                 'Maximalbestellmenge','Lagerverhalten'
                                                                );
```

Hier könnte man z.B. bei den Variablen auch noch Artikelzusatzfelder mit auslesen (Feldbezeichnungen, siehe Tabelle `artikel` in der Shop-Datenbank): `,'zusatzfeld_1', 'zusatzfeld2'`.

3.2 Bruttopreis Cache

Alle Preise, die von der Preisfindung verändert werden können (Bruttopreise), werden zwischengespeichert (Page Cache) und können im Shop über das Preisinfo-Singleton-Objekt wie folgt abgefragt werden. Diese Daten werden z.B. dann gebraucht, wenn man dem Kunden z.B. einen “Anstatt-Preis“ anzeigen will:

```
$preisinfo_obj     = preisinfo::get_instance(); // Singleton Objekt holen
$art_preisinfo_obj = $preisinfo_obj->get_art_preisinfo($artikel_id);
```

Info: Im preisfindung Objekt existiert schon ein Preisinfo Objekt: `$this->art_preisinfo_obj`.

3.3 Aktion Erkennung

Ob sich ein Artikel im Aktions-Zeitraum befindet kann ueber das Preisinfo-Objekt folgendermassen abgefragt werden:

```
if ($this->art_preisinfo_obj->get_artikel_in_aktion()) { … }
```

3.4 Beispiel Preisfindung Implementierung

```
// Unterscheidung Kunde nicht eingeloggt / eingeloggt (gut zum Testen):
if(!$this->eingeloggt){
    // Allen NICHT eingeloggten Kunden alle Artikelpreise auf 5.55 setzen
    $this->art_preis = 5.55;
}
else{
    // Allen eingeloggten Kunden alle Artikelpreise auf 2.22 setzen
    $this->art_preis = 2.22;
} // end else

// Allen (Standard PepperShop) Varianten den Preis 1.11 zuordnen
foreach($this->art_variationen_arr as $key=>$value){
    $this->art_variationen_arr[$key] = 1.11;
}

// Allen (Standard PepperShop) Optionen den Preis 3.33 zuordnen
foreach($this->art_optionen_arr as $key=>$value){
     $this->art_optionen_arr[$key] = 3.33;
}

// Ein Beispielaufruf mit Runden auf 10, sieht z.B. wie folgt aus:
$rundungsfaktor__= 10;
$waehrungsdaten  = get_waehrung(get_current_waehrungs_id());
$this->art_preis = $this->calc_gerundeter_fremdwaehrungspreis($this->art_preis,
                   $waehrungsdaten['umrechnungsfaktor'], $rundungsfaktor);
```

3.5 Session Daten

Sollte man Informationen von der aktuellen Session benötigen, kann man diese einfach einbinden:

```
global $my_session;
$sprache = $my_session->get_session_var('lang');
```

4. Lookup

4.1 Klassendefinitionen

Artikel	:{shopdir}/shop/artikel_def.php
Kategorie	:{shopdir}/shop/kategorie_def.php
Kunde	:{shopdir}/shop/kunde_def.php
Bestellung	:{shopdir}/shop/bestellung.def.php
Session	:{shopdir}/shop/session_def.php

5. Installation

5.1 Voraussetzungen

Um das Preisfindung Modul einsetzen zu können, ist ein PepperShop ab Version 3.1 (Professional oder Enterprise) erforderlich.

5.2 Dateien kopieren und hochladen

Das Preisfindung Modul besteht im Wesentlichen aus zwei Dateien, die man in seinen installierten PepperShop kopieren muss. Dazu entpackt man zuerst das mitgelieferte ZIP-Archiv. Es sind folgende Dateien enthalten, hier mit den Zielverzeichnissen dargestellt:

```
{shopdir}/shop/preisfindung.def.php                    | Datei
{shopdir}/shop/preisfindung_helper.def.php             | Datei
```

Die Dateien lassen sich einfach via FTP oder SCP zum Webserver hochladen. Der Platzhalter `{shopdir}` steht dabei für das Shopverzeichnis auf dem Webserver, wo die Datei `index.php` zusammen mit `README.txt` zu finden ist.

5.3 Modulinstallation

Nach dem Kopieren der Dateien steht das Modul unmittelbar zur Anwendung bereit.

5.4 Aktivierung / Deaktivierung / Basiskonfiguration

Wie einleitend schon erwähnt, handelt es sich bei diesem Modul um eine headless Version. Es gibt keine Web basierte Maske zur Konfiguration oder Statusanzeige (abgesehen von der Aktivierungsanzeige in Shop-Einstellungen ⇨ Shop-Konfiguration ⇨ Module).

Es gibt eine zentrale Konfigurationssteuerung für die Aktivierung und Deaktivierung dieses Moduls:

Datei: `{shopdir}/shop/config.inc.php`

```
define('PREISFINDUNG_VERWENDEN',true); // Def: false | Wenn das Preisfindungs Modul installiert ist, kann man mit dieser Konstante steuern, ob eine spezielle Preisfundung fuer Artikelpreise durchgefuehrt werden soll
```

Wenn bei der Konstantendefinition von **PREISFINDUNG_VERWENDEN** ein **false** steht, ist das Modul deaktiviert und alle Programmierungen werden nicht verwendet. Steht dort ein **true**, ist das Modul aktiv.

Mit folgender Steuerungskonstante wird allenfalls die Preisfindung für die Artikelliste ausgeschaltet. Dies sollte man sich gut überlegen. Der Defaultwert ist = `true` (eingeschaltet). Man spart lediglich Performance.

```
define('PREISFINDUNG_AUF_ERSTER_STUFE_VERWENDEN',true); // Def: true  | Wenn das Preisfindungs Modul installiert und aktiviert ist, kann man mit dieser Konstante steuern, ob die Preisfindung bei aktivierter, zweistufiger Artikeldarstellung auch auf der ersten Artikelstufe (tabellarische Uebersichtsdarstellung aller Artikel einer Kategorie) angewendet werden soll. Wenn z.B. externe Calls in ERP-Systeme anfallen, moechte man hier ev. false verwenden - dann sieht der Kunde zwar erst bei der Detailansicht 'seinen' Preis, dafuer ist die Performance ok.
```

Da die meisten Preisfindungsregeln auf Kundengruppen(-IDs) basieren, sollte man die Kundengruppen-ID (ganzzahliger, positiver, nummerischer Wert) auch für die Administration aktivieren:

```
define('KUNDENGRUPPE_ADMIN',true); // Def: false | Im Kundenmanagement gibt es die Moeglichkeit dem Kunden eine Kundengruppe zuzuordnen. Diese wird in der Datenbank gespeichert und hat sonst keine Funktionalitaet
```

6. Support

Glarotech GmbH kann Customizing Programmierungen für dieses Modul vornehmen, aber nicht für Auswirkungen selbst erstellter Programmierungen haften, resp. dafür kostenlose Supportdienstleistungen anbieten.