PEPPERSHOP

PepperShop Benachrichtigungen Modul

Anleitung

Datum
04. Oktober 2016

Version
1.4

Inhaltsverzeichnis

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. Technische Informationen

Um dieses Modul verwenden zu können, werden technisches Know-How vorausgesetzt: PHP, Cron-Jobs / geplante Tasks.

```
Kurzbeschreibung Benachritigungen / Notifications Modul:
========================================================

Was macht das Modul?
--------------------
Bei jedem Artikel, der nachbestellbar oder uneingeschränkt
verfügbar ist (Lagerverhalten) und wo der Lagerbestand = 0
ist, wird eine Info angezeigt, dass man sich informieren
lassen kann. Wenn man das als Kunde macht, wird dieser Event
registriert. Im eigenen Kundenaccount sieht der Kunde seine
noch offenen Lagerbenachrichtigungen.

Sobald im Shop der Lagerbestand des Artikels wieder > 0 wird,
verarbeitet der Shop in der Nacht auf den kommenden Tag alle
anstehenden E-Mailbenachrichtigungen und versendet dabei die
Events per Mail.


Dateien:
--------
Datei      : shop/notifications.def.php
Verzeichnis: shop/Admin/notify


**************************************************
*                                                *
* ACHTUNG: LAGERBENACHRICHTUNGSLINK WIRD NUR DANN  *
* ANGEZEIGT, WENN LAGERVERWALTUNG EINGESCHALTTET   *
* UND KUNDE_SIEHT_LAGERBESTAND EINGESCHALTET IST!  *
*                                                *
**************************************************


---
fjo, 29.09.2010
```

2. Beschreibung

- Seit dem PepperShop v.2.5 gibt es ein Modul um Benachrichtigungen zu versenden.
- Man muss einen CRON-Job (periodisch aufgerufener Task) einrichten können.
- Lagerbenachrichtigungen Ablauf:
    - Bei jedem Artikel, der nachbestellbar oder uneingeschränkt verfügbar ist (Lagerverhalten) und wo der Lagerbestand = 0 ist wird eine Info angezeigt, dass man sich informieren lassen kann. Wenn man das als Kunde macht, wird dieser Event registriert. Im eigenen Kundenaccount sieht der Kunde seine noch offenen Lagerbenachrichtigungen.
    - Sobald im Shop der Lagerbestand des Artikels wieder > 0 wird, verarbeitet der Shop in der Nacht auf den kommenden Tag alle anstehenden E-Mailbenachrichtigungen und versendet dabei die Events per Mail.

Das Modul besteht aus folgenden Dateien:

```
{shop_verzeichnis}/shop/
  - notifications.def.php                 : Kundenseitige Funktionalität
{shop_verzeichnis}/shop/Admin/notify/
    - notify.php                          : Aufruf-Datei für Cron-Job (WICHTIG!)
    - notify.def.php                      : Proceed Basisklasse
    - notify_artikel.def.php              : Konkrete Verarbeitung von Lager-Benachrichtigungen
    - notify_bewertung.def.php            : Konkrete Verarbeitung von Bewertungs-
                                            Benachrichtigungen (erst ab v.3.0)
    - notify_vorauskasse.def.php          : Konkrete Verarbeitung von Vorauskasse-Erinnerung-
                                            Benachrichtigungen (erst ab v.3.0)
```

Weiter betroffen ist die Datei `shop/kunde_account.php`, wo z.B. offene Notification Events verwaltet werden können.

3. Tabellen

Das Benachrichtigungssysytem verwendet zwei Tabellen in der Shopdatenbank:

```
notify_events
notify_liste
```

4. Cron-Job einrichten

Damit Benachrichtigungen versendet werden, muss man folgende Datei via Cron-Job einrichten:

```
{shop_verzeichnis}/shop/Admin/notify/notify.php
```

Beispiel täglicher crontab Eintrag (`crontab -l -u username`) Lagerbenachrichtigungen (Shop Webroot im Hauptverzeichnis):

```
0 5 * * * wget --http-user=shopadmin --http-password=xxxxxxxx -O notify.html
http://www.domain.tld/shop/Admin/notify/notify.php
```

Oder via CLI-Call

```
0 5 * * * cd http://www.domain.tld/shop/Admin/notify/ notify.php commandline
www.domain.tld  http://www.domain.tld
```

Manuelles Ausführen aller anliegenden Notifications: Im Browser folgende URL öffnen: http://`{domain}{shop_verzeichnis}/shop/Admin/notify/notify.php`.

5. Eventtypen

Event-ID	*Eventname*	*Verfügbar ab*	*Details*
100	Lager Benach-richtigung	v.2.5	Daten: - Verarbeitung der Events in `shop/Admin/notify/notify_artikel.def.php` - Wenn ein Artikel wieder an Lager ist, werden alle Kunden benachrichtigt, welche sich das so konfiguriert haben (E-Mail). - Insert des Events erfolgt durch Link in USER_ARTIKEL_HANDLING_AUFRUF.php der zu kunde_account.php führt (darstellen=12 mit add_... Modul-Methode). Info: Der Link wird nur angezeigt wenn in der Administration das Lagermanagement eingeschaltet ist und 'Kunde sieht Lagerbestand' auf JA konfiguriert worden ist! - Die Verarbeitung läuft grundsätzlich ohne Zeitverzögerung (also mit Granularität des Cron-Job Aufrufs - zumeist wohl täglich).
200	Bewertungs Benach-richtigung	v.3.0	Daten: - Verarbeitung der Events in `shop/Admin/notify/notify_artikel.def.php` - Wenn ein Kunde noch nicht bewertete Artikel hat, wird er darauf hingewiesen, dass er diese bewerten kann (E-Mail). - Insert des Events erfolgt nach Bestellungsabschluss in USER_BESTELLUNG_1.php, darstellen=4 ganz unten. Dort wird nur ein neuer Event eingefügt, wenn der einkaufende Kunde noch keinen nicht verarbeiteten 200-er Event hat und wenn er nicht bewertete Artikel zugeordnet hat und auch dann nur, wenn die Steuerungskonstante SEND_BEWERTUNG_EMAIL_NOTIFICATIONS = true ist (bewertung.def.php). - Die Verarbeitung läuft mit einer Verzögerung von der in der Steuerungskonstante SEND_EMAIL_NOTIFICATIONS_AFTER_DAYS in bewertung.def.php angegebenen Anzahl Tage (ausfuhren_ab Feld wird populiert) - es wird angenommen, dass der Cron-Job täglich eingestellt ist.
300	Vorauskasse Erinnerung Benach-richtigung	v.3.0	Daten: BENÖTIGT DAS BEZAHLSTATUS MODUL! - Verarbeitung der Events in `shop/Admin/notify/notify_vorauskasse.def.php` - Wenn ein Kunde seine Vorauskassebezahlung noch nicht beglichen hat (Bezahlstatus gelb/grün), so erhält er ein Erinnerungs E-Mail. - Insert des Events erfolgt nach Bestellungsabschluss in USER_BESTELLUNG_1.php, darstellen=4 ganz unten. Dort wird nur ein neuer Event eingefügt, wenn die Steuerungskonstante SEND_VORAUSKASSE_MAIL_NOTIFICATIONS_AFTER_DAYS >= 0 ist (config.inc.php). - Die Verarbeitung läuft mit einer Verzögerung von der in der Steuerungskonstante SEND_VORAUSKASSE_MAIL_NOTIFICATIONS_AFTER_DAYS angegebenen Anzahl Tage (ausfuhren_ab Feld wird populiert) - es wird angenommen, dass der Cron-Job von notify.php mindestens täglich eingestellt ist. Je nach Verwendungsart des Verarbeitungsprozesses muss man hier noch protected $pruefen_auf = 'zahlungsbest_am'; anpassen (keine Zahlungsbestätigungen, sondern nur effektive Zahlungseingänge erfassen)

6. Patches

6.1 Patch für USER_BESTELLUNG_1.php um für jeden unbewerteten Artikel einen Eintrag zu tätigen

```
// Falls das Artikelbewertungsmodul vorhanden ist und wir E-Mail Notifications für noch
// nicht
// bewertete Artikel versenden möchten, tun wir das hier
if (modul_check('bewertung') && modul_check('notifications') &&
defined('SEND_BEWERTUNG_EMAIL_NOTIFICATIONS') && SEND_BEWERTUNG_EMAIL_NOTIFICATIONS == true){
 // Bereits getätigte Artikelbewertungen auslesen:
 $bewertungen_obj = new bewertungen;
 $bewertungen_obj->get_bewertungen_eines_kunden($myKunde->Kunden_ID);
 $irrelevante_artikel = array();
 if (is_array($bewertungen_obj->bewertet_arr) && !empty($bewertungen_obj->bewertet_arr)){
   foreach($bewertungen_obj->bewertet_arr as $bewertung) {
     if (intval($bewertung->artikel_id) > 0) {
       $irrelevante_artikel[intval($bewertung->artikel_id)] = 'schon_bewertet';
     }
   }
 }

// Bereits vorhandene Notifications des Kunden auslesen:
 $notify_liste_obj = new notify_liste();
 $notify_liste_obj->get_notifications_from_kunde($myKunde->Kunden_ID,true);
 if (is_array($notify_liste_obj->notify_arr) && !empty($notify_liste_obj->notify_arr)) {
   foreach($notify_liste_obj->notify_arr as $notify_obj) {
     if (intval($notify_obj->event_id) == 200 && intval($notify_obj->value_int_1) > 0) {
       $irrelevante_artikel[intval($notify_obj->value_int_1)] = '200_event_vorhanden';
     }
   }
 }

 // Bewertungs Event einfügen, wenn Artikel noch nicht bewertet worden ist und Event noch nicht
// existiert
 // Event-ID fuer Lager-Benachrichtigungen ist 100:
 $artikel_der_bestellung = $meineBestellung->getallartikel();
 $noch_nicht_bewertete_artikel_dieser_bestellung = array();
 foreach($artikel_der_bestellung as $artikel) {
   // Falls vorhanden, die Parent-Artikel-ID verwenden:
   $artikel_id = false;
   if (defined('PARENT_SUB_ARTIKEL') && PARENT_SUB_ARTIKEL == true) {
     $artikel_id = get_parent_artikel_id($artikel->Artikel_ID,'subartikel_id');
   }
        if ($artikel_id == false) {
             $artikel_id = $artikel->Artikel_ID;
        }
        $artikel_id = intval($artikel_id);

        // Wurde dieser Artikel vom Kunden schon bewertet oder existiert der Event schon?
        if (array_key_exists($artikel_id,$irrelevante_artikel)) {
             continue;
        }
        else {
             // Pruefen, ob es sich um einen Subartikel handelt. Wenn ja, lesen wir den Parent-
             // Artikel aus und eben diesen zur Bewertung an.
 $notify_obj = new notify_request;
 if($notify_obj->insert(200,$myKunde->Kunden_ID,$artikel_id,
 $meineBestellung->Bestellungs_ID)){ // value_int_1 = Artikel-ID, value_int_2 =
 Bestellungs_ID
 insert_statistik_event(array('notification','notifications',
```

```
 'add_bewertung_notification',$myKunde->Kunden_ID,$artikel_id,
                  $meineBestellung->Bestellungs_ID));
                  $irrelevante_artikel[$artikel_id] = '200_event_vorhanden';
             }
             else{
                  // Fehlemeldung loggen und anzeigen:
                  show_error('Notification Event 200: '.£('benachrichtigung_schon_drin').
                  ' (Artikel-ID: '.intval($artikel_id).')!',false,false,true);
             }
        }
    }
    unset($artikel_der_bestellung);
}
```

6.2 Patch um Lagerbenachrichtigungen verwenden zu können, aber keine Lagerinfos anzuzeigen

- Patch erstellt für PepperShop v.2.6.0
- Datei: shop/USER_ARTIKEL_HANDLING_AUFRUF.php:

```
--- USER_ARTIKEL_HANDLING_AUFRUF.php_original   2010-09-29 15:22:04.000000000 +0200
+++ USER_ARTIKEL_HANDLING_AUFRUF.php    2010-09-29 15:32:19.000000000 +0200
@@ -1479,7 +1479,10 @@

             // Falls die Lagerverwaltung eingeschaltet ist: Lagerinformationen ausgeben
             // Die Fallunterscheidungen finden weiter oben statt, direkt unter der myarray foreach
Schleife
-            $art_tpl->set_variable('lagerinfo',htmlentities($max_kauf['lager_info']));
+// XXX>fjo: Lagerinfo nicht anzeigen - Kunde soll nur Lagerbenachrichtigung verwenden können:
+            // $art_tpl->set_variable('lagerinfo',htmlentities($max_kauf['lager_info']));
+            $art_tpl->set_variable('lagerinfo','');
+// <XXX

             // Output fuer Hidden-Felder erzeugen lassen, dabei wird die globale Artikel-Set Unique
ID ($asid) nur dann uebertragen,
             // wenn der aktuell gerenderte Artikel auch ein Artikel-Set Variationsartikel ist.
@@ -1855,6 +1858,8 @@

                 // Falls die Lagerverwaltung eingeschaltet ist: Lagerinformationen ausgeben
                 // Die Fallunterscheidungen finden weiter oben statt, direkt unter der myarray
while-list Schleife (ehemals foreach)
+// XXX>fjo: Keine Lageranzeige, obwohl so konfiguriert im Admin (wird nur fuer
Lagerbenachrichtigung verwendet!
+/*
                 if(isset($max_kauf['lager_info']) && $max_kauf['lager_info'] != '') {
                     if (defined('PARENT_SUB_ARTIKEL') && PARENT_SUB_ARTIKEL == true &&
defined('SHOW_SUBARTIKEL_INFOS_AUF_1_STUFE') && SHOW_SUBARTIKEL_INFOS_AUF_1_STUFE == true &&
intval($myarray->sublagerbestand) > 0) {
                         // (kumulierter) Lagerbestand der Subartikel darstellen
@@ -1874,6 +1879,8 @@
                     $lst_art_tpl-
>set_variable('einheit',htmlentities($einheiten['anzahl_einheit']));
                     $lst_art_tpl->parse_current_block();
                 }
+*/
+// <XXX

                 // falls promo, Promo-Indikator anzeigen
                 if ($myarray->promo == 'Y' || (defined('PARENT_SUB_ARTIKEL') && PARENT_SUB_ARTIKEL
== true && defined('SHOW_SUBARTIKEL_INFOS_AUF_1_STUFE') && SHOW_SUBARTIKEL_INFOS_AUF_1_STUFE == true
&& $myarray->subanzahlpromo > 0)) {
@@ -1967,12 +1974,17 @@
                     $lst_art_tpl->set_variable('name_label',£('artikel_name'));
                     // Lager-Ueberschrift nur ausgeben, wenn auch das Lagermgmt verwendet wird
                     $sortierfelder_weglassen = array();
+// XXX>fjo: Keine Lageranzeige, obwohl so konfiguriert (wir benötigen nur die
Lagerbenachrichtigung)
+                    $sortierfelder_weglassen[] = 'Lagerbestand';
+/*
```

```
                    if(!isset($max_kauf['lager_info']) || $max_kauf['lager_info'] == ''){
                        $sortierfelder_weglassen[] = 'Lagerbestand';
                    }
                    else {
                        $lst_art_tpl->set_variable('lager_label',£('lagerbestand'));
                    }
+*/
+// <XXX

                    // Sortierungslinks und Darstellung generieren und im Template abfuellen
                    $sortierung = create_listen_sortierung_output($sortierfelder_weglassen);
@@ -2167,6 +2179,8 @@

                // Falls die Lagerverwaltung eingeschaltet ist: Lagerinformationen ausgeben
                // Die Fallunterscheidungen finden weiter oben statt, direkt unter der myarray
while-list Schleife (ehemals foreach)
+// XXX>fjo: Keine Lageranzeige, obwohl so konfiguriert im Admin (wird nur fuer
Lagerbenachrichtigung verwendet! (einspaltige Liste)
+/*
                if(isset($max_kauf['lager_info']) && $max_kauf['lager_info'] != '') {
                    if (defined('PARENT_SUB_ARTIKEL') && PARENT_SUB_ARTIKEL == true &&
defined('SHOW_SUBARTIKEL_INFOS_AUF_1_STUFE') && SHOW_SUBARTIKEL_INFOS_AUF_1_STUFE == true &&
intval($myarray->sublagerbestand) > 0) {
                        // (kumulierter) Lagerbestand der Subartikel darstellen
@@ -2184,7 +2198,7 @@
                    $lst_art_tpl->set_variable('lagerbestand',$lagerbestand_dargestellt);
                    $lst_art_tpl-
>set_variable('einheit',htmlentities($einheiten['anzahl_einheit']));
                }
-
+*/
                // Nach dem letzten Artikel den Artikellistenfooter ausgeben
                if($artikel_in_kat == $artikel_counter){
                    // Footer-Zeile parsen
@@ -3691,4 +3705,3 @@


-
```