

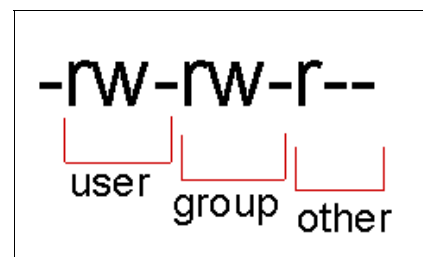
## PhPepperShop Webshopsystem

---

José Fontanil und Reto Glanzmann, Glarotech GmbH

# Installationsanleitung

*Shopinstallation ohne Telnet / SSH  
ab PhPepperShop Version 2.0.0*





## Informationen zur Anleitung

Dies ist die angepasste und erweiterte Fassung v.2.03 vom Februar 2007.

Diese Installationsanleitung kann **erst ab der PhPepperShop Version 2.0.0** verwendet werden. Für ältere PhPepperShop Versionen gibt es eine andere Anleitung.

Mit dieser Anleitung sollen alle, welche von ihrem Provider keinen Telnet oder SSH Zugang zu ihrem Webserver erhalten, den PhPepperShop installieren können.

Die Installation erfordert das manuelle Ändern von Dateien. Die Dateien des PhPepperShops sind mit UNIX-Zeilenumbrüchen geschrieben worden. Auf Windows-Rechnern sehen sie dementsprechend etwas 'komisch' aus. Wir empfehlen das Programm Proton oder PHPEd zu verwenden, dann werden die Dateien auch richtig interpretiert. Zur Not lassen sich aber alle Änderungen auch im Windows-Notepad erledigen; [Proton Editor Link](#), [PHPEd Editor Link](#).

Wir erhielten früher öfters E-Mails oder Foren-Beiträge, welche uns darauf hinweisen, dass diese Installation unnötig kompliziert ist. Wir mögen uns doch an Projekten orientieren, bei welchen man nur in einer zentralen Datei bestimmte Variablen konfigurieren muss.

Wir wollen hier nochmals betonen, dass der PhPepperShop über ein dynamisches, komfortabel zu bedienendes Installationstool verfügt, welches die komplette Shopinstallation Menü geführt abwickelt. Dieses Installationstool benötigt aber einen Telnet / SSH-Zugang!

Wer keinen Telnet / SSH-Zugang zur Verfügung hat, muss halt etwas Handarbeit über sich ergehen lassen und folgende Anleitung durch arbeiten.

## Inhaltsverzeichnis

|                                                                  |   |
|------------------------------------------------------------------|---|
| 1. Übersicht.....                                                | 3 |
| 2. Ausgangslage.....                                             | 3 |
| 3. Was benötige ich um diesen Shop zu installieren.....          | 3 |
| 4. Download des Source Codes des PhPepperShops.....              | 3 |
| 5. Editieren der anzupassenden Dateien.....                      | 3 |
| template_create.sql.....                                         | 4 |
| template_insert.sql.....                                         | 4 |
| template_del_tables_only.sql.....                                | 4 |
| initialize.php.....                                              | 4 |
| ADMIN_initialize.php.....                                        | 4 |
| ADMIN_backup.php.....                                            | 5 |
| 6. Shop Datenbank einlesen.....                                  | 5 |
| 6.1 Deinstallation.....                                          | 6 |
| 7. Hochladen des 'personalisierten' Shops auf den Webserver..... | 6 |
| 8. Anpassen der Datei-Zugriffsrechte.....                        | 6 |
| 9. Sichern der Datenbankdaten.....                               | 8 |
| 10. Erstellen des Schutzes für den Administrationsbereich.....   | 8 |
| 10.1 Vorgehen bei einer Fehlermeldung.....                       | 8 |
| 10.2 Weitere Sicherheitsaspekte.....                             | 9 |
| 11. Nachwort.....                                                | 9 |



## 1. Übersicht

Die Installation kann in folgende sechs Abschnitte unterteilt werden:

1. Herunterladen des PhPepperShop Source Codes
2. Editieren der anzupassenden Dateien
3. Hochladen der angepassten Shop-Datenbank
4. Hochladen des 'personalisierten' Shops auf den Webserver
5. Anpassen der Datei-Zugriffsrechte
6. Erstellen des Zugangschutzes für den Administrationsbereich

## 2. Ausgangslage

Es wird davon ausgegangen, dass man den Shop mit nur einem, schon existierenden Datenbankbenutzer und einer, auch schon existierenden, Datenbank installieren will. Wir wollen hier nochmals darauf hinweisen, dass unser Security Konzept mit nur einem Datenbankbenutzer nicht komplett umgesetzt werden kann. Weiter ist zu beachten, dass der schon existierende Datenbankbenutzer vielleicht auch Rechte besitzt, welche von unserem Security-Konzept her ausgeschlossen sind (GRANT, ALTER,...).

Wir gehen davon aus, dass jeder, welcher zwei dedizierte Datenbankbenutzer und eine eigene Datenbank ausschliesslich für den Shop bereitstellen kann, auch unser Installationscript per Telnet oder SSH benutzen kann, deshalb hier nur diese 'eingeschränkte' Konfiguration.

## 3. Was benötige ich um diesen Shop zu installieren

Um den PhPepperShop mit dieser Anleitung installieren zu können benötigen Sie folgende Komponenten:

- Sourcedateien des PhPepperShops (v.2.0.0 oder neuer)
- Ein ASCII-Editor wie z.B. PHPEd, Windows-Notepad, Emacs oder pico,...
- Ein FTP-Programm welches auch UNIX Dateizugriffs-Rechte ändern kann
- Einen Webaccount mit FTP-Zugang und folgender Unterstützung:
  - PhPepperShop v.2.0.0: PHP4 ab Version 4.0.6 / PHP5.0 kompatibel
  - PhPepperShop v.2.0.1: PHP4 ab Version 4.0.6 / PHP5.x kompatibel
  - GD-Library (vorzugsweise ab Version 2)
  - session.use\_trans\_sid = Off
  - allow\_call\_time\_pass\_reference = On
  - Funktion ini\_set darf nicht gesperrt sein
  - MySQL (ab v.3.23, empfohlen wird MySQL ab Version ab 4.0.4)
  - Webserver (getestet mit Apache, sollte auch auf IIS funktionieren)

## 4. Download des Source Codes des PhPepperShops

Laden Sie den PhPepperShop von folgender Internetseite herunter:

<http://www.phpeppershop.com/download.php>

Entpacken Sie die Dateien in ein Verzeichnis. Darin befindet sich jetzt das Verzeichnis `phpeppershop_src`, worin sich die Shopdateien befinden. Dieser Shop ist noch *nicht* einsetzbar. Es müssen zuerst einige Anpassungen vorgenommen werden.

## 5. Editieren der anzupassenden Dateien

Im Verzeichnis `phpeppershop_src/template/` befinden sich mehrere Dateien. Hier wollen



wir zuerst die SQL-Dateien anpassen. Diese sind für das Erstellen und Löschen der Shop-Datenbank im MySQL-Server zuständig.

### **template\_create.sql**

1. Alle Zeilen **von und mit** `{nomakeuser}` **bis und mit** `{nomakeuser}` löschen. (wären für die User-Erstellung des automatisierten Telnet/SSH-Setups `config.pl` zuständig.)
2. Alle `{shop_db}` ersetzen mit dem Namen der schon bestehenden Datenbank. ACHTUNG: Auch die geschweiften Klammern *müssen ersetzt werden!* Bsp. Aus `{shop_db}` wird dann z.B. `db1651`.

### **template\_insert.sql**

1. Alle `{shop_db}` ersetzen mit dem Namen der schon bestehenden Datenbank. ACHTUNG: Auch die geschweiften Klammern *müssen ersetzt werden!* Ersetzen wie in der Datei `template_create.sql`.

### **template\_del\_tables\_only.sql**

1. Alle Zeilen **von und mit** `{nomakeuser}` **bis und mit** `{nomakeuser}` löschen. (wären im automatisierten Telnet/SSH-Setup für das Löschen der User zuständig.)

### **initialize.php**

Nun muss zuerst das Verzeichnis gewechselt werden: Im Verzeichnis `phpeppershop_src/shop` findet man die nächste anzupassende Datei: `initialize.php`.

In dieser Datei muss man auf der zweitletzten Zeile (?> nicht mit eingerechnet) die Datenbankbindung anpassen. Per Default wird hier folgender Eintrag verwendet:

```
$Database = new TMySQLDatabase('{hostname}', '{shop_db}', '{shopuser}', '{shopuserpwd}');
```

Anstatt `{hostname}` soll man den Rechnernamen angeben, auf welchem MySQL läuft (z.B. `localhost` oder `database.meinedomain.de`). Den Bezeichner `{shop_db}` wird durch den Namen der schon existierenden Datenbank ersetzt. Zum Schluss noch `{shopuser}` durch den Namen des schon bestehenden Datenbank-Users ersetzen. Mit `{shopuserpwd}` ist das Passwort des Datenbank-Users gemeint. Bsp.:

```
$Database = new TMySQLDatabase('localhost', 'db1651', 'db1651_user', 'pfefferschote');
```

Die geschweiften Klammern müssen also auch ersetzt werden aber die Hochkommata nicht!

### **ADMIN\_initialize.php**

Dieselben Änderungen muss man an der Datei `ADMIN_initialize.php` vornehmen. Diese liegt im Verzeichnis `phpeppershop_src/shop/Admin`.

Auch hier muss man auch die zweitletzte Zeile abändern (?> nicht mitgezählt):

```
$Admin_Database = new TMySQLDatabase('{hostname}', '{shop_db}', '{shopadmin}', '{shopadminpwd}');
```

Anstatt `{hostname}` soll man den Rechnernamen angeben, auf welchem MySQL läuft (z.B. `localhost` oder `database.meinedomain.de`). Den Bezeichner `{shop_db}` wird durch den Namen der schon existierenden Datenbank ersetzt. Zum Schluss noch `{shopadmin}` durch den Namen des schon bestehenden Datenbank-Users ersetzen. Mit `{shopadminpwd}` ist das Passwort des Datenbank-Users gemeint.

Diese 'doppelte' Arbeit ist nötig, weil der PhPepperShop für den Administrator einen anderen



Datenbankuser benutzen *kann*. Wir gehen hier aber davon aus, dass man nur einen Datenbankbenutzer hat. Bsp.:

```
$Admin_Database = new TMySQLDatabase('localhost', 'db1651', 'db1651_user', 'pfefferschote');
```

## ADMIN\_backup.php

Die letzte anzupassende Datei ist die Datei ADMIN\_backup.php, welche auch im Verzeichnis phpeppershop\_src/shop/Admin liegt. Hier muss man auch eine gesonderte Datenbankanbindung konfigurieren. Die zu verändernden Zeilen findet man im oberen Teil des Quellcodes, nachdem man etwas nach unten gescrollt hat.

```
$dbhost='{hostname}';  
$dbuser='{shopadmin}';  
$dbpass='{shopadminpwd}';  
$dbname='{shop_db}';
```

Damit man jemandem im Geschäft die Erlaubnis geben kann Backups durchzuführen, wurde das Backup-Management wieder mit einer Möglichkeit versehen eigene Datenbankverbindungsdaten zu definieren. Achtung: In dieser Datei ist die Reihenfolge der Variablen leicht anders als vorher.

Anstatt {hostname} soll man den Rechnernamen angeben, auf welchem MySQL läuft (z.B. localhost oder database.meinedomain.de). Den Bezeichner {shop\_db} wird durch den Namen der schon existierenden Datenbank ersetzt. Zum Schluss noch {shopadmin} durch den Namen des schon bestehenden Datenbank-Users ersetzen. Mit {shopadminpwd} ist das Passwort des Datenbank-Users gemeint.

Bsp.

```
$dbhost="localhost";  
$dbuser="db1651_user";  
$dbpass="pfefferschote";  
$dbname="db1651";
```

## 6. Shop Datenbank einlesen

Mit einem Hilfsprogramm wie z.B. phpMyAdmin oder einem anderen Programm, kann man nun die beiden Dateien template\_create.sql und template\_insert.sql (Verzeichnis phpeppershop\_src/template/) in die Datenbank einlesen. Hier eine Kurzanleitung für das Programm *phpMyAdmin*, <http://www.phpmyadmin.net> (Der Provider sollte phpMyAdmin oder ein ähnliches Datenbankzugriffsprogramm anbieten).

HINWEIS: Bevor Sie die Dateien einlesen möchten wir darauf hinweisen, dass die in den Dateien vorhandenen Befehle nur *einmal* ausgeführt werden können. Bitte nicht zweimal auf OK klicken. Wenn es wider erwarten doch zu Fehlermeldungen kommen sollte, muss man zuerst die Datei template\_del\_tables\_only.sql laufen lassen, bevor man es nochmals neu mit template\_create.sql und dann template\_insert.sql versuchen kann.

Für phpMyAdmin Versionen ab 2.7.0 geht das Einlesen der SQLs wie folgt vor:

[http://www.phpeppershop.com/phpeppershop\\_files/sql\\_insert\\_help\\_ab\\_phpMyAdmin\\_2\\_7\\_0.gif](http://www.phpeppershop.com/phpeppershop_files/sql_insert_help_ab_phpMyAdmin_2_7_0.gif)

phpMyAdmin ab v.2.7.0:

1. Links die eigene Datenbank anwählen
2. 'importieren' Reiter (oben) auswählen
3. 'Durchsuchen' anklicken (dieser Knopf kann auch anders benannt sein)
4. Datei auswählen (zuerst template\_create.sql, dann template\_insert.sql)
5. Falls angezeigt, jetzt noch die 'Zeichenkodierung der Datei' angeben: **latin1**, dies ist ganz wichtig. Wenn der Standard (utf8) verwendet wird, funktioniert der Shop nicht korrekt! Wie



schon erwähnt, ist diese Einstellung erst im Zusammenhang mit einer MySQL-Datenbank ab Version 4.1.x und einer aktuellen phpMyAdmin Version vorhanden.

6. Jeweils mit Öffnen bestätigen
7. SQLs ausführen -> jeweils mit OK bestätigen

phpMyAdmin <= 2.3.0:

Für phpMyAdmin Versionen bis v.2.3.0 haben wir auch eine grafische Version der Anleitung: [http://phpeppershop.streaming-24.com/phpeppershop\\_files/sql\\_insert\\_help\\_neu.gif](http://phpeppershop.streaming-24.com/phpeppershop_files/sql_insert_help_neu.gif)

8. Links die eigene Datenbank anwählen
9. SQL Reiter (oben mitte) auswählen
10. Browse auswählen
11. Datei auswählen (zuerst template\_create.sql, dann template\_insert.sql)
12. Jeweils mit OK bestätigen

Ältere phpMyAdmin Versionen (kann zu Problemen führen -> siehe Forum):

In grafischer Form für alte phpMyAdmins findet man diese Schritte auch hier: [http://phpeppershop.streaming-24.com/phpeppershop\\_files/sql\\_insert\\_help.gif](http://phpeppershop.streaming-24.com/phpeppershop_files/sql_insert_help.gif)

1. Links die eigene Datenbank anwählen
2. SQL-Befehl(e) ausführen in Datenbank... oder Datei
3. Datei auswählen (zuerst template\_create.sql, dann template\_insert.sql)
4. Auf OK klicken

Falls hier von der Datenbank Fehler vorkommen, soll man nochmals alle vom Shop erzeugten Tabellen in der Datenbank löschen und die SQL-Skripte in Stückchen einzulesen versuchen. Es kann sein, dass es zu einem PHP-Timeout gekommen ist, wenn das Einlesen zulange gedauert hat (> 30s meistens). Hilfe bietet hier auch unsere Foren auf der PhPepperShop Webseite <http://www.phpeppershop.com/>. Eine Liste der Tabellen findet man in der Datei:

phpeppershop\_src/templates/template\_del\_tables\_only.sql.

Nachdem die Datenbank nun erfolgreich hochgeladen wurde und die Shopdaten angepasst sind, können wir die Shopdateien auf den Webserver laden und zum Punkt 7 weitergehen.

## 6.1 Deinstallation

Wenn man den Shop wieder deinstallieren will, so muss man auf dem gleichen Weg die Datei `template_del_tables_only.sql` ausführen. Dabei werden alle vom PhPepperShop verwendeten Tabellen gelöscht. Achtung: *ALLE* Daten sind verloren, wenn nicht ein Backup erstellt und gesichert wurde!

## 7. Hochladen des 'personalisierten' Shops auf den Webserver

Um den Shop auf den Webserver zu laden, sollte man ein *FTP-Programm* benutzen. Bei mir hat sich CuteFTP bewährt, es geht aber auch jedes andere. Leechftp hat sich als besonders robust erwiesen. Hier hat aber jeder seine eigene Vorliebe. Man sollte das ganze Verzeichnis `phpeppershop_src` komplett hochladen. Man kann es (sollte es) natürlich auch umbenennen (z.B. `shop`). Der Name, des Verzeichnisses ist später in der URL ersichtlich. Bsp.  
<http://www.meinserver.de/shop/>

## 8. Anpassen der Datei-Zugriffsrechte

Wir können nicht genug betonen, dass dieser Punkt penibel genau gelesen und durchgeführt werden muss. Viele Anfragen im Forum haben sich erledigt, nachdem die Benutzer nochmals



diesen Punkt der Anleitung genau durchgearbeitet haben. Der PhPepperShop wird **NICHT** korrekt funktionieren, wenn diese Schritte nicht verstanden / abgearbeitet werden!

Grund: Da der PhPepperShop selbstständig einige Operationen auf Dateien ausführt, müssen wir auf Servern mit folgenden Merkmalen mehrere Datei- und Verzeichnisrechte entsprechend anpassen, so dass auch der Benutzer des Webservers sie schreibend verändern darf:

Systemmerkmale von betroffenen Systemen (weit verbreitet):

- Der Webserver ist ein UNIX System - z.B. Linux, MacOS X, FreeBSD, OpenBSD, Solaris, ...
- PHP läuft als Webservermodul

Falls nicht bekannt ist, ob der Webserver dieser Konfiguration entspricht, kann man den Provider anfragen, er muss es wissen und wird sicher bereitwillig Auskunft geben.

Falls PHP auf dem Webserver nicht als Webservermodul läuft, sondern als *CGI-Programm* aufgerufen wird, dürfen folgende Änderungen in diesem Kapitel **NICHT** durchgeführt werden. Man kann dann die Warnungen getrost ignorieren und *direkt zu Kapitel 9* gehen. Dasselbe gilt für Windows basierte Webserver, da Windows keine UNIX Dateizugriffsrechte besitzt.

Wenn die oben genannten Systemmerkmale aber auf den verwendeten Webserver zutreffen, so verhindert diese Konfiguration leider das Verändern / Erstellen von Dateien über den PhPepperShop-Administrationsbereich - der PhPepperShop kann somit also gar nicht korrekt funktionieren...

Schreiten wir zur Tat: Man muss nun dem hochgeladenen Ordner, in welchem sich der Shop befindet und allen sich darin befindlichen Unterverzeichnissen das *UNIX Dateizugriffsrecht 755* geben, allen Dateien *644* (dies sollte standardmässig schon der Fall sein). Das macht man am einfachsten bequem über ein FTP Programm (z.B. das kostenlose FileZilla).

Einige spezielle PhPepperShop Unterverzeichnisse und Dateien benötigen jedoch aus oben erwähnten Gründen andere Zugriffsrechte. Folgende Verzeichnisse benötigen jeweils *777*, allen Dateien hingegen muss man *666* vergeben. Damit werden die Dateien (u.a.) für den Webserver Benutzer beschreibbar:

```
phpeppershop_src/shop/shopstyles.css
phpeppershop_src/shop/Bilder/*           (Alle Dateien in diesem Unterverzeichnis)
phpeppershop_src/shop/Buttons/*          (Alle Dateien in diesem Unterverzeichnis)
phpeppershop_src/shop/ProdukteBilder/*   (Das Verzeichnis und die Dateien darin!)
phpeppershop_src/shop/ProdukteDetails/*  (Das Verzeichnis und die Dateien darin!)
phpeppershop_src/shop/Admin/Backups      (Das Verzeichnis Backups selbst ist hier gemeint)
phpeppershop_src/shop/Admin/Import       (Das Verzeichnis Import selbst ist hier gemeint)
phpeppershop_src/shop/Admin/log          (Das Verzeichnis log selbst ist hier gemeint)
phpeppershop_src/shop/Admin/import_template (Das Verzeichnis import_template selbst
und die Datei import_beschreibung.tpl.html sind hier gemeint)
phpeppershop_src/shop/language/html_templates (Das Verzeichnis und alle Dateien darin)
⇒ phpeppershop_src/shop/language/de/localetext.inc.php
⇒ phpeppershop_src/shop/language/de/html_templates/* (Das Verzeichnis und alle Dateien darin)
⇒ phpeppershop_src/shop/language/de/text_templates/* (Das Verzeichnis und alle Dateien darin)
⇒ phpeppershop_src/shop/language/de/Buttons/*       (Alle Dateien in diesem Verzeichnis)
⇒ phpeppershop_src/shop/language/en/localetext.inc.php
```



- ⇒ `phpeppershop_src/shop/language/en/html_templates/*` (Das Verzeichnis und alle Dateien darin)
- ⇒ `phpeppershop_src/shop/language/en/text_templates/*` (Das Verzeichnis und alle Dateien darin)
- ⇒ `phpeppershop_src/shop/language/en/Buttons/*` (Alle Dateien in diesem Verzeichnis)
  
- ⇒ `phpeppershop_src/shop/language/fr/localetext.inc.php`
- ⇒ `phpeppershop_src/shop/language/fr/html_templates/*` (Das Verzeichnis und alle Dateien darin)
- ⇒ `phpeppershop_src/shop/language/fr/text_templates/*` (Das Verzeichnis und alle Dateien darin)
- ⇒ `phpeppershop_src/shop/language/fr/Buttons/*` (Alle Dateien in diesem Verzeichnis)
  
- ⇒ `phpeppershop_src/shop/language/it/localetext.inc.php`
- ⇒ `phpeppershop_src/shop/language/it/html_templates/*` (Das Verzeichnis und alle Dateien darin)
- ⇒ `phpeppershop_src/shop/language/it/text_templates/*` (Das Verzeichnis und alle Dateien darin)
- ⇒ `phpeppershop_src/shop/language/it/Buttons/*` (Alle Dateien in diesem Verzeichnis)
  
- ⇒ `phpeppershop_src/shop/language/rm/localetext.inc.php`
- ⇒ `phpeppershop_src/shop/language/rm/html_templates/*` (Das Verzeichnis und alle Dateien darin)
- ⇒ `phpeppershop_src/shop/language/rm/text_templates/*` (Das Verzeichnis und alle Dateien darin)
- ⇒ `phpeppershop_src/shop/language/rm/Buttons/*` (Alle Dateien in diesem Verzeichnis)

! Hier wurden die standardmässig mitgelieferten Sprachverzeichnisse angegeben. Die Dateiattribute der drei Verzeichnisse `html_templates`, `text_templates` und `Buttons` (und ihrer Dateien) müssen aber auch für jede weitere installierte, hier nicht erwähnte Sprache, entsprechend bearbeitet und gesetzt werden.

**Nochmals:** Diese Anpassungen sind sehr wichtig. *Der Shop wird nicht wie gewünscht funktionieren, wenn diese Änderungen nicht alle korrekt abgearbeitet wurden!* (CGI siehe oben)

## 9. Sichern der Datenbankdaten

Man muss dem Verzeichnis `phpeppershop_src/template` das Datei-zugriffsrecht `700` gegeben werden. Allen Dateien darin, muss `600` zugeordnet werden. Dies, damit kein unbefugter die Datenbankangaben sehen kann. Als Alternative kann man dieses Unter-verzeichnis auch gar nicht erst auf den Webserver laden.

## 10. Erstellen des Schutzes für den Administrationsbereich

Damit der Administrationsbereich nicht von jedermann betreten werden kann, muss er Passwortgeschützt werden. Dies machen wir mit einer `.htaccess` und `.htpasswd` Datei, eine 'Funktionalität' des Apache-Webservers. Wenn ein Microsoft® IIS oder sonst ein Server eingesetzt wird, muss man sich bezüglich des Verzeichniszugangschutzes mit dem Provider in Verbindung setzen.

Heute ist es üblich, dass der Provider ein Tool zur Verfügung stellt, mit welchem man selbst einen Verzeichniszugriffsschutz für ein beliebiges Verzeichnis im eigenen Webaccount erstellen und verwalten kann. Man kann damit nun einen aktivierten Schutz für das Verzeichnis `phpeppershop_src/shop/Admin` erstellen.

Für den Fall, dass der Provider kein Tool dieser Art zur Verfügung stellt, kann das beigelegte Tool `SHOP_HTACCESS.php` verwendet werden um einen Schutz zu erstellen:

*Temporär* geben wir dem Verzeichnis `phpeppershop_src/shop/Admin` das Zugriffsrecht `777`.

Wir öffnen in unserem Browser die folgende Datei:

`heraufgeladenes_shopverzeichnis/shop/Admin/SHOP_HTACCESS.php`

Achtung: Gross-/Kleinschreibung beachten. Hier können wir nun unseren `.htaccess`-Schutz generieren lassen. Nachdem dies abgearbeitet ist, **muss** man die Zugriffsrechte des Admin-Verzeichnisses wieder auf `755` zurücksetzen. Unter Windows gibt es keine UNIX-Zugriffsrechte – einfach ignorieren.



## 10.1 Vorgehen bei einer Fehlermeldung

Wenn man versucht hat einen htaccess-Schutz einzurichten und man erhält danach eine '500 Internal Server Error' Fehlermeldung, so hat die Erstellung des .htaccess-Schutzes nicht wie gewünscht funktioniert. Meistens liegt es daran, dass man einen Advanced-htaccess-Schutz benutzen will, der Webserver diese Funktionalität aber nicht unterstützt. Wie dem auch sei. Um den fehlerhaften .htaccess-Schutz zu löschen, soll man einfach im Admin-Verzeichnis die beiden Dateien .htaccess und .htpasswd löschen. Der Punkt vor dem Dateinamen bedeutet, dass die Dateien als 'versteckt' angesehen werden. Wenn man sie also im FTP-Tool nicht sieht, sollte man nach entsprechender Einstellung suchen.

## 10.2 Weitere Sicherheitsaspekte

Die .htpasswd Datei sollte nicht in einem per Web zugänglichen Verzeichnis liegen. Man sollte diese Datei ausserhalb des Webverzeichnisses ablegen und den Pfad in der .htaccess Datei entsprechend anpassen. Per Default liegt sie im Ordner: `upgloadetes_shopverzeichnis/shop/Admin/`.

Weiter wollen wir darauf hinweisen. Dass .htaccess im Basic-Mode die Passworteingabe unverschlüsselt und deshalb unsicher entgegen nimmt. Mit der Advanced Einstellung (entspricht der Digest Methodik) können aber viele Webserver (noch) nicht umgehen.

## Der Shop sollte jetzt einsatzbereit sein!

Wenn noch ungeklärte Fragen vorhanden sein sollten, lesen Sie bitte zuerst die FAQs durch und schauen Sie danach im Forum vorbei. (<http://www.phpeppershop.com>)

## 11. Nachwort

Man sollte nun als erstes im Administrationsmenü seines PhPepperShops zum Punkt '*Shop Konfiguration ansehen*' gehen (auf den Stern im oberen Teil der Seite klicken). Hier sollte man die Diagnoseausgaben anschauen. Rote Punkte deuten auf Probleme hin. Weitere Infos findet man dann in den PhPepperShop Foren.

Danach sollte man in der Shopadministration folgende Menüs der Reihe nach durchgehen um die Shopkonfiguration an die eigenen Bedürfnisse anzupassen:

- *Allgemeine Shopeinstellungen* (besonders wichtig: Adress- und E-Mail Angaben)
- Lieferländer und Versandarten
- Währungseinstellungen
- Sprachen
- Layout Management, Shopbuttons (Knöpfe) und Bilder (Hintergrund & Shoplogo) hochladen