



# **Installationsanleitung**

Shop-Installation ohne SSH  
ab PepperShop Version 5.0

## **Datum**

04. November 2016

## **Version**

5.2

## Informationen zur Anleitung

Dies ist die angepasste und erweiterte Fassung dieser Anleitung vom Juni 2014.

Diese Installationsanleitung kann **erst ab der PepperShop Version 5.0** verwendet werden. Für ältere PepperShop Versionen gibt es andere Anleitungen (siehe Anleitungen im Bereich "Support" - "Anleitungen" auf [www.peppershop.com](http://www.peppershop.com)).

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

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

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, oder die einen Webinstaller anbieten. Wir wollen hier nochmals betonen, dass der PepperShop über ein dynamisches, komfortabel zu bedienendes [Installationstool](#) verfügt, welches die komplette Shop-Installation Menü geführt abwickelt. Dieses Installationstool benötigt aber einen Telnet oder SSH-Zugang. Wer keinen Telnet / SSH-Zugang zur Verfügung hat, muss etwas Handarbeit über sich ergehen lassen und folgende Anleitung einmal durch arbeiten.

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 GmbH  
Toggenburgerstrasse 156  
CH-9500 Wil

[info@glarotech.ch](mailto:info@glarotech.ch)  
Tel. +41 (0)71 923 08 58  
[www.glarotech.ch](http://www.glarotech.ch)

## Inhaltsverzeichnis

1. Übersicht.....	4
2. Ausgangslage.....	4
3. Was benötige ich um diesen Shop zu installieren.....	4
4. Download des Source Codes des PepperShops.....	4
5. Editieren der anzupassenden Dateien.....	4
template_create.sql.....	5
template_insert.sql.....	5
template_del_tables_only.sql.....	5
initialize.php.....	5
ADMIN_initialize.php.....	5
ADMIN_backup.php.....	5
6. Shop Datenbank einlesen.....	6
6.1 Deinstallation.....	7
7. Hochladen des 'personalisierten' Shops auf den Webserver.....	7
8. Anpassen der Datei-Zugriffsrechte.....	7
9. Sichern der Datenbankdaten.....	8
10. Daten aufräumen mithilfe eines Cron-Jobs.....	8
11. Erstellen des Schutzes für den Administrationsbereich.....	8
11.1 Vorgehen bei einer Fehlermeldung.....	8
11.2 Weitere Sicherheitsaspekte.....	9
12. Erste Schritte.....	9

## 1. Übersicht

Die Installation kann in folgende sechs Abschnitte unterteilt werden:

1. Herunterladen des PepperShop 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 Zugangsschutzes 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 Installationsscript per SSH benutzen kann, deshalb hier nur diese 'eingeschränkte' Konfiguration.

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

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

- Source-Dateien des PepperShops (v.5.0 oder neuer)
- Ein Text-Editor wie z.B. [Notepad++](#), PHPEd, Sublime, Windows-Notepad, vi, Emacs oder nano, ...
- Ein FTP-Programm welches auch UNIX Dateizugriffs-Rechte ändern kann
- Einen Webaccount mit FTP-Zugang und folgender Unterstützung:
  - PHP5 ab Version 5.4.0 (ältere Versionen sind nicht mehr kompatibel!)
  - GD-Library (ab Version 2)
  - Die PHP-Funktion `ini_set` darf nicht gesperrt sein
- MySQL ab Version ab 5.1.23
- Webserver (Apache, IIS, ...)

## 4. Download des Source Codes des PepperShops

Laden Sie den PepperShop in der aktuellen Version über Ihrem Account herunter. Öffnen Sie das zu folgende Webseite und geben Sie Ihre Login-Daten in der Box auf der linken Seite ein.

PepperShop Verkaufssystem: <http://www.peppershop.com/shop/>

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

## 5. Editieren der anzupassenden Dateien

Im Verzeichnis `peppershop_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 `peppershop_src/shop` findet man die nächste anzupassende Datei: `initialize.php`.

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

```
$Database = new $db_klasse('{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 $db_klasse('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 `peppershop_src/shop/Admin`.

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

```
$Admin_Database = new $db_klasse('{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 PepperShop für den Administrator einen anderen Datenbankuser benutzen *kann*. Wir gehen hier aber davon aus, dass man nur einen Datenbankbenutzer hat. Bsp.:

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

## ADMIN\_backup.php

Die letzte anzupassende Datei ist die Datei `ADMIN_backup.php`, welche auch im Verzeichnis `peppershop_src/shop/` **Admin** liegt. Hier muss man auch eine gesonderte Datenbank-anbindung konfigurieren. Die zu verändernden Zeilen findet man im oberen Teil des Quellcodes, nachdem man etwas nach unten gescrollt hat (dahinter gibt es noch Kommentare, die hier nicht abgebildet sind).

```
$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 Datenbankanbindungsdaten 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 `peppershop_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.

Verwenden Sie das DB-Tool Ihres Providers. Für phpMyAdmin geht das Einlesen der SQLs wie folgt vor: [http://www.peppershop.com/peppershop\\_files/sql\\_insert\\_help\\_ab\\_phpMyAdmin\\_2015.png](http://www.peppershop.com/peppershop_files/sql_insert_help_ab_phpMyAdmin_2015.png)

1. Links die eigene Datenbank anwählen
2. 'importieren' Reiter (oben) auswählen
3. 'Durchsuchen' anklicken (dieser Knopf kann auch anders benannt sein)
4. Datei auswählen (Im Ordner `template`: zuerst `template_create.sql` wählen)
5. Falls angezeigt, jetzt noch die 'Zeichenkodierung der Datei' prüfen: **UTF-8**, dies ist ganz wichtig. Wenn z.B. latin1 verwendet wird, *funktioniert der Shop nicht korrekt!*
6. Jeweils mit Öffnen bestätigen
7. Nun `template_create.sql` SQL ausführen -> dann mit OK bestätigen
8. Jetzt die neu erzeugte Datenbank wählen und hier via importieren das zweite SQL-File namens `template_insert.sql` ausführen

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 PepperShop Website <http://www.peppershop.com/>. Eine Liste der Tabellen findet man in der Datei:

```
peppershop_src/templates/template_del_tables_only.sql.
```

Nachdem die Datenbank nun erfolgreich hochgeladen wurde und die Shopdaten angepasst sind, können wir die Shop-Dateien 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 PepperShop 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. Ein kostenloses, robustes FTP-Programm ist der FileZilla Client: <http://filezilla-project.org>. Hier hat aber jeder seine eigene Vorliebe... Man sollte das ganze Verzeichnis `peppershop_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 PepperShop wird **NICHT** korrekt funktionieren, wenn diese Schritte nicht verstanden / abgearbeitet werden!

Grund: Da der PepperShop 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 Webserver sie schreibend verändern darf:

Systemmerkmale von betroffenen Systemen (weit verbreitet):

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

Falls nicht bekannt ist, ob der Webserver dieser Konfiguration entspricht, kann man den Provider anfragen, er wird dies wissen und kann sicher schnell und kompetent 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 PepperShop-Administrationsbereich - der PepperShop kann somit also gar nicht korrekt funktionieren...

Schreiten wir zur Tat: Entweder man ruft im Browser folgende Datei auf, oder man macht sich auf den Weg alle Anpassungen manuell durchzuführen. Das Automatik-Script befindet sich am unten angegebenen Ort. Es arbeitet mit FTP und fragt nach den Zugangsdaten:

heraufgeladenes\_shopverzeichnis/shop/Admin/shop\_berechtigungen\_clean.php

**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 `peppershop_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 Unterverzeichnis auch gar nicht erst auf den Webserver laden.

## 10. Daten aufräumen mithilfe eines Cron-Jobs

Für den PepperShop sollte ein periodisch gestarteter Aufruf eingerichtet werden, der eine Aufräum Funktion aktiviert. Dabei wird unter anderem die Datenbank optimiert und nicht mehr benötigte Cache-einträge geleert.

HTTP[S]-Aufruf auf folgende Datei einrichten:

```
http[s]://{www.ihredomain.tld}/{webshop_verzeichnis}/shop/Admin/garbage_collection.php
```

Bei UNIX/Linux Systemen kann man z.B. einen Cron-Job mit folgendem Kommando verwenden:

```
wget http[s]://{www.ihredomain.tld}/{webshop_verzeichnis}/shop/Admin/garbage_collection.php
```

Wenn die Administration mit einem Passwort geschützt ist, kann man den Benutzernamen und das Passwort wie folgt angeben:

```
http[s]://benutzername:passwort@{www.ihredomain.tld}/...
```

## 11. 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 Verzeichnisschutzes 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 `peppershop_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:

Falls der Shop als Webserver-Modul unter UNIX/Linux läuft, geben wir dem Verzeichnis `peppershop_src/shop/Admin` *temporär* 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.



### 11.1 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.

### 11.2 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:  
heraufgeladenes\_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.peppershop.com>)

## Bitte führen Sie nun die ersten Schritte aus.

## 12. Erste Schritte

Der Shop steht bereit, jetzt geht es an die Konfiguration und die Erklärungen, wie man mit Bestellungen umgeht. Am besten lesen Sie das Dokument 'Erste Schritte', welches auf viele Fragen von neuen PepperShop Administratoren eingeht.

### Detailierte Anleitung der weiteren Schritte

[http://www.peppershop.com/peppershop\\_files/erste\\_schritte.pdf](http://www.peppershop.com/peppershop_files/erste_schritte.pdf)

**!** Bei Fragen rund um den PepperShop hilft oft ein Klick auf das Hilfearchiv links im des Hauptmenü Ihrer PepperShop-Administration. Weitere Hilfe findet man in den umfangreichen PepperShop Foren auf <http://www.peppershop.com>, oder individuell via [Support-Paket](#).