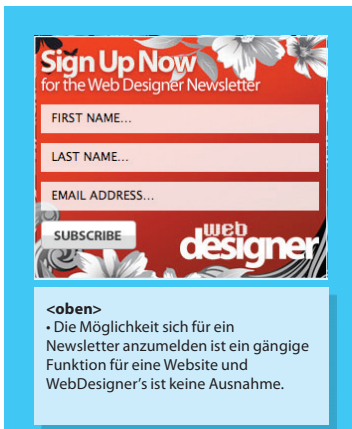
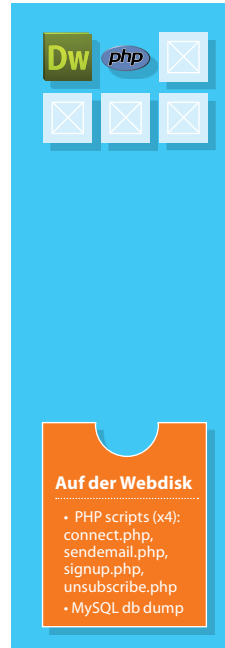


# Entwicklung eines Newsletter-Scripts in PHP und MySQL

Newsletter sind ein hervorragendes Mittel um mit Kunden und Usern in Kontakt zu bleiben

**Tools | Tech | Trends** Dreamweaver, PHPMyAdmin (or SQL interface of your choice) **Experte** Sam Hampton-Smith



Viele Unternehmen im Netz leben davon, dass sie für ihre Kunden die Verwaltung und den Versand von Newsletter (auch eShots oder eFlyers genannt) übernehmen. Für viele Großfirmen ist es durchaus ratsam, diesen Kernteil der Kommunikation zu den Kunden auszulagern und sich auf das zu konzentrieren, was sie am besten können. Diese Dienstleistung kann jedoch schnell sehr teuer werden. Dabei ist es für Websites kleineren Umfangs sehr einfach, selbst die Newsletter-Verwaltung in die Hand zu nehmen.

In diesem Tutorial werden wir über alles sprechen, was Sie für den Anfang brauchen: Wir werden ein Registrierungs-Script entwerfen, das die Emailadressen der Abonnenten sammelt und es erlaubt, sich von dem Newsletter abzumelden. Beispielhaft werden wir aufzeigen, wie sich eine Email an alle Abonnenten im Kollektiv versenden lässt. Es kommt PHP und MYSQL zur Anwendung, aber keine Sorge, wenn Sie keines von beidem kennen, durch die vielen Einzelschritte, wird es für Sie dennoch leicht zu verstehen sein.

Wir werden einen Überblick über die besten Methoden geben, die Ihnen helfen nicht als Spamverteiler eingestuft zu werden. Auch gehen wir auf Sicherheitsaspekte ein, die Sie beachten sollten, wenn Sie Input von den Besuchern Ihrer Website sammeln. Am Ende wird Ihnen die Grundstruktur eines Management-Werkzeugs für Kundenkontakte zur Verfügung stehen, das Sie an Ihre Bedürfnisse und die Ihrer Website anpassen können.

Wir werden ein Newsletter-Registrierungs-Script entwerfen, das Emailadressen von Benutzern sammelt und ihnen das Abmelden ermöglicht.

## 01 Ein erstes PHP-Script

Wir werden unsere Emailnewsletter-Anwendung in PHP programmieren. Stellen Sie also sicher, dass Ihr Hostler PHP und MySQL-Datenbanken zulässt. Falls Sie auf Ihrem eigenen Rechner entwickeln wollen, empfehlen sich MAMP (für Macs) und WAMP (für Windows PCs) für Ihren lokalen Server. Näheres finden Sie auf [mamp.org](http://mamp.org) und [wampserver.com](http://wampserver.com).

## 02 Datenbank erstellen

Öffnen Sie PHPMyAdmin entweder mithilfe der MAMP/WAMP-Steuerung oder den Vorgaben Ihres Hosters. Erstellen Sie eine neue Datenbank (es sei denn es existiert bereits eine) und legen Sie einen neuen Benutzer mit den entsprechenden Rechten für diese Datenbank an. Bei Unsicherheit fragen Sie Ihren Hostler um Hilfe.

## 03 Datenbanktabellen

Wir werden eine Tabelle für die Benutzernamen und Emailadressen verwenden. Zur besseren Übersichtlichkeit erhält jeder Datenbankeintrag eine automatisch ansteigende ID (auch wenn dies für dieses Tutorial nicht notwendig ist, empfiehlt es sich jeder Zeile einen eindeutigen Wert zuzuordnen).

```
001 CREATE TABLE `subscribers` (
002 `id` int(11) NOT NULL AUTO_INCREMENT,
003 `name` varchar(255) NOT NULL,
004 `email` varchar(255) NOT NULL,
005 PRIMARY KEY (`id`)
006 ) ENGINE=MyISAM DEFAULT CHARSET=latin1
AUTO_INCREMENT=5 ;
```

## 04 Das Registrierungsformular

Wir werden ein Standard-HTML-Formular einsetzen. Ohne einen größeren Umfang an PHP-Prozesslogik geht es jedoch nicht. Daher empfiehlt es sich, diese Seite komplett eigenständig aufzusetzen, um sie später in andere Seiten einzufügen. Beginnen Sie mit einem einfachen Formular, das nach Namen und Emailadresse fragt.

```
001 <form action="<?php echo $_SERVER['PHP_
002 SELF']; ?>" method="post">
003 <fieldset>
004 <legend>Subscribe to Our Newsletter
005 &nbsp;&nbsp;&nbsp;</legend>
006 <?php if ($feedback!="") echo('<p>!' . $feedback . '</
007 p>'); ?>
008 <label>Name: <input name="name" type="text"
009 /></label>
010 <label>Email: <input name="email" type="text"
011 /></label>
012 <label><input type="submit" value="Sign
013 Up!" /></label>
014 </fieldset>
015 </form>
```

## 05 Fortsetzung des Registrierungsformulars

Nach dem wir nun alles nachgeprüft haben kann es immer noch sein, dass die Einträge des Benutzers ungültig sind. Für

**<oben>**

- Laden Sie MAMP oder WAMP, um die Anwendung auf Ihrem eigenen Rechner laufen zu lassen. Bei einem Testserver fragen Sie Ihren Hostler nach dem Zugang für die Datenbank.
- Mit PHPMyAdmin oder einer vergleichbaren Software richten Sie eine Datenbank samt Benutzer ein.
- In Ihrer Datenbank erstellen Sie eine Tabelle mit Feldern für eine ID, dem Namen und die Emailadresse. Die ID sollte eine automatisch zunehmende Zahl sein.
- Beugen Sie SQL-Angriffen vor, indem Sie die vom Benutzer übermittelten Strings komplett säubern, bevor eine Datenbankanfrage gestartet wird.

dieses ist es jedoch ärgerlich, wenn alle Daten erneut eingegeben werden müssen. Stattdessen füllen wir die Formularfelder automatisch vor, sodass der Benutzer nur noch den Fehler korrigieren muss und erneut absenden kann.

```
001 <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
002 <fieldset>
003 <legend>Subscribe to Our Newsletter
004 <?php if ($feedback!="") echo("<p>.$feedback.</p>"); ?>
005 <label>Name: <input name="name" type="text" value="<?php echo $name; ?>" /></label>
006 <label>Email: <input name="email" type="text" value="<?php echo $email; ?>" /></label>
007 <label><input type="submit" value="Sign Up!" /></label>
008 </fieldset>
009 </form>
```

### 06 Datenüberprüfung

Nach Absenden des Formulars, ist es nötig die Daten auf ihre Gültigkeit zu überprüfen. Am meisten leuchtet natürlich ein, dass wir prüfen müssen, ob die Emailadresse echt ist. Dazu müssen wir eine Reihe von Tests mit der übermittelten Adresse durchführen.

```
001 // Set up a variable for error messages
- make it empty
002 $feedback = "";
003 if (!$email) {
004 // No email address was provided
```

```
005 $feedback .= '<strong>Please enter your email address</strong><br />';
006 }
007 if (!$name) {
008 // No name provided
009 $feedback .= '<strong>Please enter your name</strong><br />';
010 }
```

### 07 Gültige Email-Domain?

PHP kann für einen inversen DNS-Lookup verwendet werden. Dabei werden Webeinträge untersucht, um festzustellen, ob eine bestimmte Domain existiert und ob sie bestimmte Anfragen verarbeiten kann. Mithilfe dieser Funktionalität prüfen wir ob die Domain der verwendeten Emailadresse gültig ist.

```
001 // First, we'll see if the domain is valid for email
002 list($userName, $mailDomain) = split("@", $email);
003 if (!checkdnsrr($mailDomain, "MX")) {
004 // this email domain is not valid
005 $feedback .= '<strong>Invalid email domain</strong><br />';
006 }
```

### 08 Weitere Tests

Nun haben wir herausgefunden, dass eine gültige Domain benutzt wird. Wir sollten jedoch auch prüfen, ob die Adresse das allgemeingültige Standardformat besitzt. Das heißt: Ist ein @ vorhanden, wurden nur gültige Zeichen verwendet. Zum Glück geschieht das auf einfache Weise mit einem Standardausdruck, der den String-Input prüft.

```
001 if(!pregi("^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$"; $email)) {
002 // This email address doesn't seem valid
003 $feedback .= '<strong>Your email address doesn't appear to be valid - please check and try again';
004 }
```

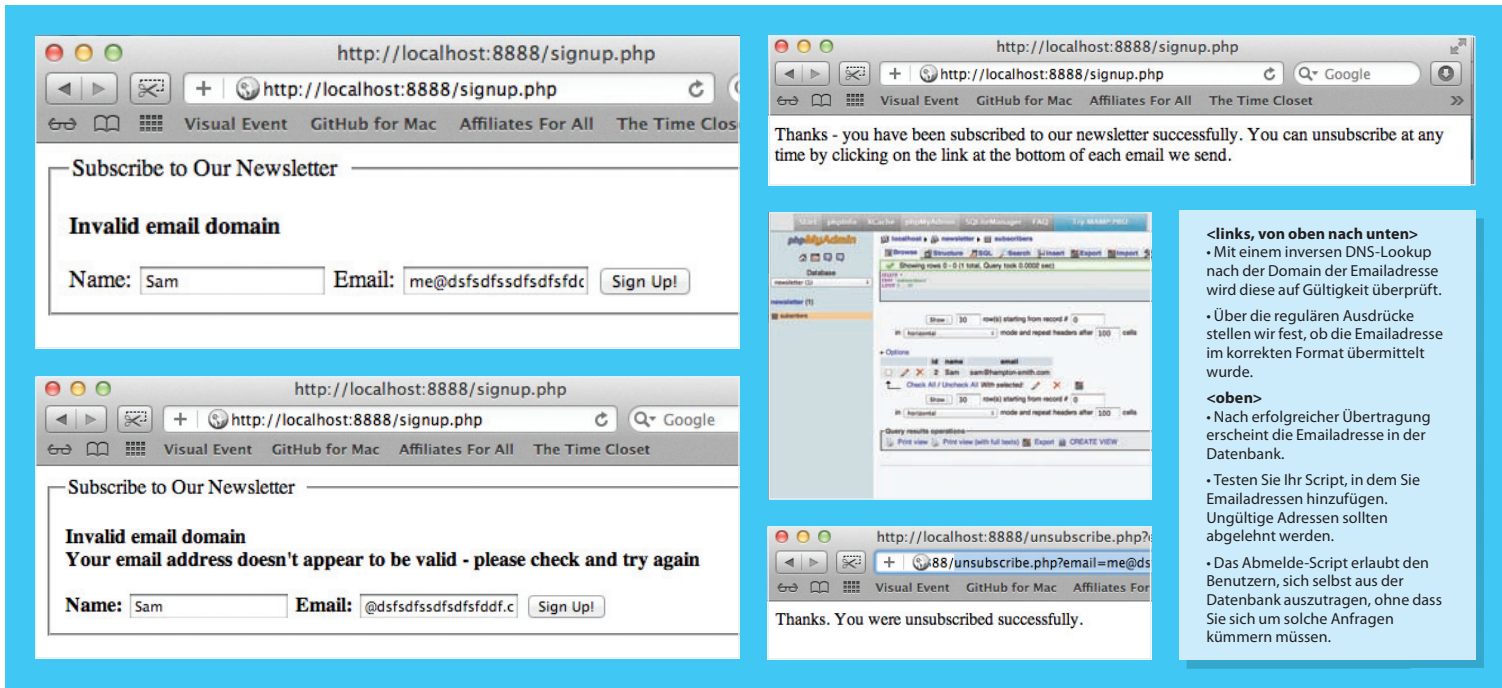
### 09 Angriffe per SQL-Injection

Bestimmte ASCII-Zeichen können dazu verwendet werden, Abfragen in PHP und MySQL zu verfälschen. Diese werden standardmäßig durch die Datenbankengine als Input gewertet und erlauben so börsartigen Hackern mithilfe dieser Zeichen Ihr Script zu sprengen. Im schlimmsten Fall könnten Sie, abhängig vom Code, Einblick in alle Benutzerdaten geben oder Zugang zu Ihrem Server gewähren. Wir werden Maßnahmen treffen, um dies zu verhindern. Lesen Sie mehr dazu in der Dokumentation auf Seite 44.

```
001 // This function attempts to loosely clean the input to prevent SQL injection attacks
002 function cleaninput($value, $DB) {
003 if (get_magic_quotes_gpc()) {
004 $value = stripslashes( $value );
005 }
006 return mysql_real_escape_string( $value, $DB );
007 }
```

### 10 Die Verbindung zur Datenbank

Es sind jedoch einige Kniffe nötig, bevor unsere PHP-Seite mit der MySQL-Datenbank kommunizieren kann und Daten extrahieren, sowie einfügen und löschen kann.



**<links, von oben nach unten>**

- Mit einem inversen DNS-Lookup nach der Domain der Emailadresse wird diese auf Gültigkeit überprüft.
- Über die regulären Ausdrücke stellen wir fest, ob die Emailadresse im korrekten Format übermittelt wurde.

**<oben>**

- Nach erfolgreicher Übertragung erscheint die Emailadresse in der Datenbank.
- Testen Sie Ihr Script, in dem Sie Emailadressen hinzufügen. Ungültige Adressen sollten abgelehnt werden.
- Das Abmelde-Script erlaubt den Benutzern, sich selbst aus der Datenbank auszutragen, ohne dass Sie sich um solche Anfragen kümmern müssen.

**Reguläre Ausdrücke**

Ein regulärer Ausdruck ist ein System, mit dessen Hilfe geprüft werden kann, ob eine Zeichenkette bestimmten Regeln entspricht. Es können z. B. Emailadressen auf ihre Gültigkeit untersucht werden – siehe auch [www.regular-expressions.info](http://www.regular-expressions.info).

Mehrere MySQL-Funktionen sind in PHP bereits integriert. Das macht es sehr einfach, denn jetzt brauchen Sie nur noch die Adresse Ihrer Datenbank, den Namen, den Benutzernamen und das Passwort. (Daten, die Ihnen Ihr Host mitteilen wird, bzw von WAMP/MAMP während des Setups genannt werden.)

```
001 include_once "connect.php";
002 Contents of connect.php:
003 <?php
004 // Be sure to update the values below with your
    own settings
005 // These will be provided by your host
006 $host = "localhost";
007 $username = "news";
008 $pass = "password";
009 $db = "newsletter";
010 $myconnection = mysql_connect($host,
    $username,$pass) or die ("could not connect to
    mysql");
011 mysql_select_db($db) or die ("Couldn't
    connect to the database");
012 ?>
```

**11 Bestehende Einträge**

Bevor die Daten des Besuchers in der Datenbank gespeichert werden, muss geprüft werden, ob sie nicht schon vorhanden sind. Andernfalls würde dieser den Newsletter mehrfach erhalten. Sollte die Adresse bereits vorliegen, ist der Datenbankeintrag nicht leer und eine Fehlermeldung kann ausgegeben werden.

```
001 $sql = mysql_query("SELECT * FROM subscribers
```

```
WHERE email='{$email}");
002 $numRows = mysql_num_rows($sql);
003 if ($numRows>0) {
004 // The email address provided is already in the
    database
005 $feedback = '<strong>That email address is
    already subscribed.</strong>';
006 }
```

**12 Datenbankeintrag einfügen**

Mit einem INSERT Statement gelangt die Information des Besuchers in unsere Datenbank. Die Daten sind bereits „gesäubert“, um uns vor SQL-Injection-Angriffen zu schützen. (Aus Platzgründen konnten wir bisher nur einen kleinen Überblick zu Scriptsicherheit geben. Es lohnt sich aber hierzu weiter Nachforschungen zu betreiben, um so Ihre Scripts vor Hackerangriffen zu schützen.) Nun setzen wir unser SQL-Query zusammen und führen es aus.

```
001 $insertresult = mysql_query("INSERT INTO
    subscribers (name, email) VALUES('{$name'},{$mai
    l}')") or die (mysql_error());
002 // If insert was successful, mark the completed
    flag as true
003 if ($insertresult) {
004     $completed = true;
005 }
```

**13 Dankesnachricht**

Da der Besucher nun erfolgreich in unserer Datenbank eingetragen ist, können wir mit einer Dankesnachricht bestätigen, dass er sich erfolgreich registriert hat. Man könnte mit einigen Code-Anpassungen in den letz-

ten Schritten dieses Tutorials genauso eine Email aussenden, um das Abonnement des Newsletters zu bestätigen.

```
001 // Show the form if the
    $completed flag is f else
002 if
    ($completed==false) {
003 ?>
004 <form action="<?php echo $
    SERVER['PHP_ SELF']; ?>" method="post">
005 <fieldset>
006 <legend>Subscribe to Our Newsletter
    &nbsp;&nbsp;&nbsp;</legend>
007 <?php if ($feedback!="") echo('<p>!.feedback.</
    p>'); ?>
008 <label>Name: <input name="name" type="text"
    value="<?php echo $name; ?>" /></label>
009 <label>Email: <input name="email" type="text"
    value="<?php echo $email; ?>" /></label>
010 <label><input type="submit" value="Sign
    Up!" /></label>
011 </fieldset>
012 </form>
013 <?php
014 } else {
015 // Otherwise, show a thank you message
016 echo("Thanks - you have been subscribed
    to our newsletter successfully. You can unsubscri
    be at any time by clicking on the link at the bottom of
```