

Emails aus PHP mit PHPMailer versenden

Oft ist es notwendig, dass PHP Scripte Emails verschicken sei es für Kontaktformulare, die Benachrichtigung bei bestimmten Ereignissen oder für das Versenden von Zugangsdaten oder ähnlichen Informationen.

PHP bietet mit der Funktion [mail\(\)](#) eine entsprechende Möglichkeit. Diese ist jedoch sehr spartanisch und nur für die einfachsten Anwendungsfälle ausreichend.

Mit einem ansehnlichen Funktionsumfang wie die Unterstützung von HTML-Mails mit alternativer Textversion, dem Versand über SMTP-Server mit Authentifizierung oder der Unterstützung für eingebettete Bilder schafft die auf [SourceFourge](#) erhältliche Klasse [PHPMailer](#) Abhilfe.

Installation von PHPMailer

Die Installation von phpmailer gestaltet sich denkbar einfach. Nach dem Download der aktuellen Version wird dieser auf der Festplatte entpackt. Anschließend werden die beiden PHP-Dateien class.phpmailer.php und class.smtp.php sowie das Verzeichnis languages auf den Webpace geladen – in unserem Fall in ein Verzeichnis namens phpmailer:

Erste Schritte

phpmailer ist jetzt einsatzfähig und kann in eigenen Scripten verwendet werden. Dazu wird die Klasse zuerst mit `require()` eingebunden und anschließend wird unter dem Namen `$mail` mit `new()` eine Instanz der Klasse gebildet, über die später Emails verschickt werden können:

```
<?
    require('phpmailer/class.phpmailer.php');

    $mail = new PHPMailer();
?>
```

Wenn dieses Script erfolgreich auf dem Server aufgeführt wird, kann die erste Testmail verschickt werden. Dazu müssen PHPMailer einige Informationen wie der Name und die Adresse des Absenders, die Adresse des Empfängers, Betreff und Text der Email mitgeteilt werden:

```
<?
    //Klasse einbinden
    require('phpmailer/class.phpmailer.php');

    //Instanz von PHPMailer bilden
    $mail = new PHPMailer();

    //Absenderadresse der Email setzen
    $mail->From = "test@tanmar.info";

    //Name des Absenders setzen
    $mail->FromName = "TanMar Tutorials";

    //Empfängeradresse setzen
    $mail->AddAddress("test@tanmar.de");

    //Betreff der Email setzen
```

```

$mail->Subject = "Die erste Mail";

//Text der EMail setzen
$mail->Body = "Hallo! \n\n Dies ist die erste Email mit PHPMailer!";

//EMail senden und überprüfen ob sie versandt wurde
if(!$mail->Send())
{
    // $mail->Send() liefert FALSE zurück: Es ist ein Fehler aufgetreten
    echo "Die Email konnte nicht gesendet werden";
    echo "Fehler: " . $mail->ErrorInfo;
}
else
{
    // $mail->Send() liefert TRUE zurück: Die Email ist unterwegs
    echo "Die Email wurde versandt.";
}
?>

```

Passen Sie die Adressen an und testen Sie das Script auf Ihrem Webservice. PHPMailer nutzt mir diesen Einstellungen die mail() Funktion von PHP zum Versand. Die Email wird dabei als reine Textmail versandt.

Emails an mehrere Empfänger senden

Emails gleichzeitig an mehrere Empfänger senden

Die Funktion AddAddress() kann auch mehrfach aufgerufen werden, um eine Email gleichzeitig an mehrere Empfänger zu senden:

```

//Mehrere Empfänger setzen
$mail->AddAddress('test1@tanmar.de', 'Testempfänger');
$mail->AddAddress('test2@tanmar.de');

```

Wie das Beispiel zeigt, ist es auch möglich, der Funktion zusätzlich zur Email Adresse den Namen des Empfängers zu übergeben. Bei dieser Art des Versands können die Empfänger die Email Adressen der anderen Empfänger sehen.

Weiterhin ist es möglich, zusätzliche Empfänger als CC oder BCC hinzuzufügen:

```

//Empfänger einer Kopie setzen
$mail->AddBCC('test1@tanmar.de',);
//Empfänger einer Blindkopie setzen
$mail->AddCC('test2@tanmar.de');

```

Emails separat an mehrere Empfänger senden

Wenn jeder Empfänger seine "eigene" Email bekommen soll, muss die Liste der Empfänger in PHPMailer nach dem Versand einer Email zurückgesetzt werden, so dass jede Email nur an einen Empfänger geht. Dies ist über die Funktion ClearAddresses() möglich:

```

//Liste der Empfänger

```

```

$receivers = array("test1@tanmar.de", "test2@tanmar.de");

//Liste der Empfänger durchlaufen
foreach ($receivers as $receiver)
{
    //Empfänger setzen
    $mail->AddAddress($receiver);

    //Email senden und überprüfen ob sie versandt wurde
    if(!$mail->Send())
    {
        // $mail->Send() liefert FALSE zurück: Es ist ein Fehler aufgetreten
        echo "Die Email konnte nicht gesendet werden<br>";
        echo "Fehler: " . $mail->ErrorInfo;
    }
    else
    {
        // $mail->Send() liefert TRUE zurück: Die Email ist unterwegs
        echo "Die Email wurde versandt.<br>";
    }

    //Empfänger zurücksetzen
    $mail->ClearAddresses();
}

```

Diese Konstellation eignet sich beispielsweise ideal für den Versand eines Newsletters. Da der Versand von vielen Emails relativ zeitaufwändig werden kann, empfiehlt es sich eventuell, vorab die Zeitbegrenzung über die PHP-Funktion [set_time_limit\(\)](#) zu erhöhen. Dies ist jedoch nur möglich, wenn der [Safe Mode](#) in PHP deaktiviert ist.

Versand von HTML Emails

PHPMailer unterstützt auch den Versand von Emails im HTML-Format. Dabei wird zusätzlich zu der Nachricht im HTML-Format eine Alternativnachricht im Textformat mit in die Email eingefügt. Die Wahl des Formats erfolgt dabei über folgenden Befehl:

```

$mail->IsHTML(true); //Versand im HTML-Format festlegen

```

Die beiden Nachrichtenteile werden dabei wie folgt gesetzt:

```

//Text der Email setzen
$email_text = "Hallo! \n\n <b>Dies ist die erste Email mit
PHPMailer!</b>";

//Body enthält die HTML-Nachricht
//Zeilenumbrüche werden in <br> umgewandelt
$mail->Body = nl2br($email_text);

//Altbody enthält die alternative Textform
//mit strip_tags() werden die HTML-Tags entfernt
$mail->AltBody = strip_tags($email_text);

```

Versand von Emails mit Attachments

PHPMailer unterstützt auch den Versand von Emails mit Anhängen. Dabei können sowohl

Dateien, die auf dem Server liegen als auch Text aus Variablen als Anhang zu einer Email hinzugefügt werden:

```
//Eine Datei vom Server als Attachment anhängen
$mail->AddAttachment('./datei.zip','anhang.zip');

//Text aus einer Variable als Attachment hinzufügen
$mail->AddStringAttachment($email_text,'anhang.htm');
```

Der zweite Parameter gibt dabei jeweils an, unter welchem Namen der Anhang zu der Email hinzugefügt werden soll. Dieser muss nicht unbedingt mit dem Namen auf dem Server übereinstimmen.

Falls über eine Instanz von PHPMailer Emails mit unterschiedlichen Attachments verschickt werden sollen, so können die hinzugefügten Attachments mit folgendem Befehl zurückgesetzt werden:

```
$mail->ClearAttachments(); //Attachments entfernen
```

Versand über einen SMTP-Server

Falls notwendig kann PHPMailer Emails auch über einen beliebigen SMTP-Server verschicken. In unseren Tests funktionierte dies ohne Probleme In Verbindung mit gängigen Email-Servern wie Sendmail, Postfix, QMail und sogar mit Microsoft Exchange.

Um den Versand über einen SMTP-Server zu konfigurieren sind lediglich zwei Befehle notwendig:

```
$mail->IsSMTP(); //Versand über SMTP festlegen
$mail->Host = "mail.tanmar.de"; //SMTP-Server setzen
```

Viele SMTP-Server verlangen zwischenzeitlich eine Authentifizierung. Diese Methode wird auch als "SMTP Auth" bezeichnet. Dabei muss zum Versand von Emails ein Benutzername und in Passwort übergeben werden. Die entsprechenden Befehle hierfür lauten wir folgt:

```
$mail->SMTPAuth = true; //Authentifizierung aktivieren
$mail->Username = "benutzername"; // SMTP Benutzername
$mail->Password = "geheim"; // SMTP Passwort
```

Ansonsten unterscheidet sich die Handhabung von PHPMailer beim Versand über einen SMTP-Server nicht von den vorangegangenen Beispielen.

Zeichensatz festlegen

Wenn mehrsprachige Emails versandt werden sollen bzw. die Sprache der Webseite einen bestimmten Zeichensatz erfordert, kann dieser auch in PHPMailer festgelegt werden:

```
$mail->CharSet = "utf-8"; //UTF-8 Kodierung festlegen
```

In der Regel wird sich bei mehrsprachigen Projekten der UTF-8 Zeichensatz anbieten. Der Inhalt des versandten Textes muss natürlich im entsprechenden Zeichensatz kodiert sein.