

Mails versenden mit Mail::Sender

Warum eigentlich Mail::Sender?

Es ist praktisch, ein Modul zum Versenden von Mails zu verwenden, da man so viel Arbeit spart. So ist es eher umständlich einen Anhang mit sendmail zu verschicken (<http://wiki.perl-community.de/bin/view/Wissensbasis/SendmailAttachments>). Oder HTML-Mails verschicken – mit Mail::Sender ganz einfach.

Mit diesem Modul kann man Mails über beliebige SMTP-Server verschicken. Bei den meisten braucht man jedoch einen Usernamen und ein Passwort, die man aber zu 99% hat, weil man bei einem FreeMail-Anbieter seinen Account hat oder das Postfach des eigenen Webhosters benutzt. Mail::Sender bietet die Möglichkeit der Authentifikation.

Wie verschicke ich denn jetzt Mails mit dem Modul?

Gehen wir jetzt mal von einer einfachen Mail – also plain-text und ohne Anhang – aus. Dann sieht das ganze so aus:

```
my $mailtext = "Das ist der Text\nden ich gerne senden wuerde...";

my $sender = new Mail::Sender {
    smtp => 'smtp.isp.tld' ,
    from => 'test@your-domain.tld' ,
    auth => 'LOGIN' ,
    authid => 'username' ,
    authpwd => 'password',
    on_errors => undef,
} or die "Can't create the Mail::Sender object: $Mail::Sender::Error\n";

$sender->MailMsg({ to => 'empfaenger@domain.tld',
    subject => 'Testmail',
    msg => $mailtext }) or print $Mail::Sender::Error;Sender::Error;
```

Mit `$sender = new Mail::Sender` wird ein neues Objekt erzeugt. Diesem übergibt man ein paar Parameter zur Konfiguration – welcher SMTP-Server soll benutzt werden, wer ist der Absender usw.

Mit `$sender->MailMsg()` wird dann die Mail geöffnet. Dort legt man dann auch fest, an wen die Mail gehen soll, was der Betreff ist und der Mailtext wird übergeben. Und schon ist die Mail gesendet.

Wie verschicke ich HTML-Mails?

Das ist auch nicht sonderlich schwierig.

```
my $form = '<h1>Test</h1><center>HTML-Mails sind toll</center>' ;
my $sender = new Mail::Sender {
    smtp => 'smtp.isp.tld' ,
    from => 'test@your-domain.tld' ,
    auth => 'LOGIN' ,
    authid => 'username' ,
    authpwd => 'password',
    on_errors => undef,
} or die "Can't create the Mail::Sender object: $Mail::Sender::Error\n";

$sender->OpenMultipart({to => 'empfaenger@domain.tld' ,
    subject => "HTML test ",
    ctype => "text/html",
    encoding => "7bit"}) or die $Mail::Sender::Error,"n";
$sender->SendEx($form);
$sender->Close();
```

Man legt einfach den HTML-Quelltext fest. Dann erzeugt man wieder ein Objekt mit der Konfiguration und diesmal öffnet man die Mail mit `$sender->OpenMultipart()`. Dort wird dann auch festgelegt, dass es eine HTML-Mail werden soll über `ctype => 'text/html'`. Über die Methode `SendEx()` wird dann der HTML-Text in die Mail eingefügt. Das `Close()` bewirkt das Senden der Mail.

Und wie kann ich eine Datei anhängen?

Dafür gibt es die Methode `Attach()`! Man übergibt der Methode einfach den Namen der Datei bzw. eine Liste der Dateinamen. Das sieht dann so aus:

```
my $file = 'fad/zur/zusendenen.datei' ;
my $form = '<h1>Test</h1><center>HTML-Mails sind toll</center>' ;
my $sender = new Mail::Sender {
    smtp => 'smtp.isp.tld' ,
    from => 'test@your-domain.tld' ,
    auth => 'LOGIN' ,
    authid => 'username' ,
    authpwd => 'password',
} or die "Can't create the Mail::Sender object: $Mail::Sender::Error\n";

$sender->OpenMultipart({to => 'empfaenger@domain.tld' ,
    subject => "HTML test ",
    ctype => "text/html",
    encoding => "7bit"}) or die $Mail::Sender::Error,"n";
$sender->SendEx($form);
$sender->Attach($file);
$sender->Close();
```

Natürlich geht das auch in Plain-text-Mails:

```
$sender = new Mail::Sender {
    smtp => 'mail.yourISP.com' ,
    from => ' somebody@somewhere.com' ,
    on_errors => undef,
}
    or die "Can't create the Mail::Sender object: $Mail::Sender::Error\n";
$sender->Open({
    to => ' mama@home.org, papa@work.com',
    cc => ' somebody@somewhere.com' ,
    subject => ' Sorry, I' llcome later.'
})
    or die "Can't open the message: $sender->{'error_msg' }\n";
$sender->SendEnc($text);
$sender->Attach($file);
$sender->Close() or die "Failed to send the message: $sender->{'error_msg' }\n"{'error_msg' }
\n";
```

Weiterführende Literatur:

<http://search.cpan.org/~jenda/Mail-Sender-0.8.10/Sender.pm> – dort sind noch jede Menge weitere Möglichkeiten aufgeführt!