Sending HTML e-mail using Email::Stuffer

Email::Stuffer Email::Sender::Transport::SMTP

Prev Next

There are many ways to send e-mail using Perl. Email::Stuffer provides a very simple API for using some of the well written e-mail sending modules.

Download: mp4 webm

YouTube

Sending simple text e-mail

```
use strict;
use warnings;

use Email::Stuffer;

my $text = <<"END";
Dear Perl Maven user,

thank you for subscribing to the Perl Maven Pro!
END

Email::Stuffer
    ->text_body($text)
    ->subject('Hello')
    ->from('Gabor Szabo <gabor@perlmaven.com>')
    ->to('Foo <foo@perlmaven.com>')
    ->send;
```

First we use a here document to create the content of the e-mail.

Then, instead of using the `new` method, Email::Stuffer allows us to create an object using a number of methods. For example, the `text_body` method will receive the content of the e-mail, create an object and return it.

On this object we can call other methods. For example the `subject` method allows us to set the subject of the e-mail. This too will return the object so we can stack the method-calls on each-other Finally calling the `send` method will actually send the e-mail.

In order for this to work one needs to have some kind of a sendmail clone installed on the local machine.

Using SMTP

If you don't have that, no problem, you can use the SMTP server of your ISP:

With Email::Suffer we can use any of the Email::Sender transport layers. We'll use Email::Sender::Transport::SMTP.

After loading the module we can add a call to the `transport` method of Email::Stuffer, and pass an `Email::Sender::Transport::SMTP` object to it that was created using the parameter `host` that holds the name of the SMTP server.

```
use strict;
use warnings;

use Email::Stuffer;
use Email::Sender::Transport::SMTP ();

my $text = <<"END";
Dear Perl Maven user,

thank you for subscribing to the Perl Maven Pro!
END

Email::Stuffer
    ->text_body($text)
    ->subject('Hello')
    ->from('Gabor Szabo <gabor@perlmaven.com>')
    ->transport(Email::Sender::Transport::SMTP->new({
        host => 'mail.perlmaven.com',
    }))
    ->to('Foo <foo@perlmaven.com>')
    ->send;
```

This will already work even if the local machine does not have a working sendmail command.

Sending HTML mail

If we would like to make our message fancier and have an HTML version as well, it is very easy. We just create the HTML version of the message, in this example using a here-document, and then we include it in the e-mail message using the `html_body` method. The e-mail will have both a text and an HTML content.

```perl
use strict;
use warnings;

use Email::Stuffer;
use Email::Sender::Transport::SMTP ();

my $text = <<"END";
Dear Perl Maven user,

thank you for subscribing to the Perl Maven Pro!
END

my $html = <<"END";
<h1>Hi</h1>
<a href="https://perlmaven.com/">Perl Maven</a>
END

Email::Stuffer
    ->text_body($text)
    ->html_body($html)
    ->subject('Hello')
    ->from('Gabor Szabo <gabor@perlmaven.com>')
    ->transport(Email::Sender::Transport::SMTP->new({
        host => 'mail.perlmaven.com',
    }))
    ->to('Foo <foo@perlmaven.com>')

    ->send;
```

Sending to multiple addresses

If we would like to send the same message to multiple addresses we can add all the addresses to the `to` field, but then every recipient will see the addresses of all the other people. Probably you don't want that.

We could recreate the whole Email::Stuffer object again and again, but that is a bit of a waste.

We can also create an Email::Stuffer object from the common values (text_body, html_body, subject, from), store it in a variable (`$email`) and then use that variable to send the e-mail to each recipient separately:

```perl
use strict;
use warnings;

use Email::Stuffer;
use Email::Sender::Transport::SMTP ();

my $text = <<"END";
Dear Perl Maven user,

thank you for subscribing to the Perl Maven Pro!
END

my $html = <<"END";
<h1>Hi</h1>
<a href="https://perlmaven.com/">Perl Maven</a>
END
my $email = Email::Stuffer
    ->text_body($text)
    ->html_body($html)
    ->subject('Hello')
    ->from('Gabor Szabo <gabor@perlmaven.com>')
    ->transport(Email::Sender::Transport::SMTP->new({
        host => 'mail.perlmaven.com',
    }));

my @addresses = ('Foo <foo@perlmaven.com>', 'Bar <bar@perlmaven.com>');
foreach my $addr (@addresses) {
    $email->to($addr)
          ->send;
}
```