

```
#!/usr/bin/env perl

use strict;
use warnings;
use v5.10;
use File::HomeDir;
use PDF::API2;
use Path::Class::Dir ();
use Mojolicious::Lite;
use File::Temp qw( tempdir );
use POSIX qw( strftime );

plugin 'TtRenderer';
plugin 'RenderFile';

my $pdf_directory = Path::Class::Dir->new( File::HomeDir->my_home, 'PDF' );
my $tmp_directory = Path::Class::Dir->new( tempdir( CLEANUP => 1 ) );

get '/' => sub {
    my($self) = @_;
    $self->redirect_to('pdf_list');
} => 'index';

get '/pdf' => sub {
    my($self) = @_;

    my $id = 0;
    my @pdfs = map { my $name = $_->basename; $name =~ s/\.pdf$//; { name => $name, id => $id++,
page_count => PDF::API2->open($_)->pages } }
        grep { (not $_->is_dir) and ($_->basename =~ /\.pdf$/) }
        $pdf_directory->children( no_hidden => 1 );

    $self->render( pdfs => \@pdfs );
} => 'pdf_list';

get '/pdf/#filename.pdf' => sub {
    my($self) = @_;
    my $filename = $pdf_directory->file($self->param('filename') . '.pdf');
    if(-f $filename && -r $filename)
    {
        $self->render_file(filepath => $filename);
    }
    else
    {
        $self->render_not_found;
    }
} => 'pdf_download';

post '/pdf/merge' => sub {
    my($self) = @_;
    my @names = map { $self->param($_ . "_name") } grep /^pdf_\d+$/, $self->param;
    if(@names > 0)
    {
        $self->redirect_to('pdf_merge', pdf_list => join('/', @names) );
    }
    else
    {
        $self->flash(error => 'No PDFs specified for merge');
        $self->redirect_to('pdf_list');
    }
} => 'pdf_merge_form';

get '/pdf/merge/*pdf_list' => sub {
```

```

my($self) = @_;
my @source_pdf_list = split '/', $self->param('pdf_list');

my $dest_pdf = PDF::API2->new;

foreach my $source_pdf_filename (map { $pdf_directory->file("$_.pdf") } grep !/^\.\/, @source_pdf_list)
{
    my $source_pdf = PDF::API2->open($source_pdf_filename);
    foreach my $page (1..$source_pdf->pages)
    {
        $dest_pdf->importpage($source_pdf, $page, $dest_pdf->page);
    }
}

my $timestamp = strftime "%Y%m%d%H%M", localtime time;

# TODO, these temp files get cleaned up when the script exits
# (if it does so nicely), but it would be better to cleanup the
# file after the transaction instead.
my $temp_file = File::Temp->new(TEMPLATE => "pdf_merge_${timestamp}_XXXX", SUFFIX => '.pdf',
DIR => $tmp_directory, UNLINK => 0 );
binmode $temp_file;
print $temp_file $dest_pdf->stringify;
close $temp_file;

$self->render_file(filepath => $temp_file->filename);

} => 'pdf_merge';

@ARGV = ('daemon', -l => 'http://127.0.0.1:3001') if @ARGV == 0 && !$ENV{TEST_PDF_MERGE};
app->secret(rand);
app->start;

=head1 NAME

pdf-merge - Web interface for merging PDF documents.

=head1 SYNOPSIS

% pdf-merge

then point your browser to

=over 4

=item * http://localhost:3001

=back

=head1 DESCRIPTION

This application provides a web interface for merging multiple
PDF documents so they can be sent to a printer for physical
copies.

My apartment doesn't have room for a printer (not with all my
junk which is apparently more important), so when I need to print
something I use a virtual PDF printer (cups-pdf in Debian, other
distributions and operating systems probably have similar packages)
and take the PDFs to FedEx Office. Unfortunately the self service
facility for printing multiple PDFs has a sucky interface so it
is better to go there with one big PDF rather than many little

```

PDFs. The intent is to use the virtual PDF printer to print to ~/PDF, and then merge them with this web application into a single PDF which I save to the memory stick which I take to the FedEx Office.

The application is implemeted with Mojolicious, plus a few other modules available on CPAN.

By default, pdf-merge only listens to port 3001 (not to conflict with other Mojolicious applications) and binds only to 127.0.0.1. If you want to bind to something else start the application with the normal Mojolicious start up

```
% pdf-merge daemon -l http://*:3002
[Wed Aug  8 22:22:23 2012] [info] Listening at "http://*:3002".
Server available at http://127.0.0.1:3002.
```

=head1 SEE ALSO

```
L<Mojolicious>, L<Mojolicious::Plugin::TtRenderer>,
L<Mojolicious::Plugin::RenderFile>
```

=head1 AUTHOR

Graham Ollis <plicease@wdlabs.com>

=head1 COPYRIGHT

Copyright 2012 Graham Ollis

=head1 LICENSE

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

=cut

\_\_DATA\_\_

@@ pdf\_list.html.tt

```
[%
  WRAPPER layouts/default.html.tt
  title = 'PDF'
%]
<form method="post" action="[% h.url_for('pdf_merge_form') %]">
  <input type="submit" value="merge" />
  [% FOREACH pdf IN pdfs %]
    <input type="hidden" name="pdf_[% pdf.id %]_name" value="[% pdf.name %]" />
  [% END %]

  <table border="1">

    <tr>
      <th>include</th>
      <th>name</th>
      <th>pages</th>
    </tr>

    [% FOREACH pdf IN pdfs %]

      <tr>
        <td><input type="checkbox" checked=checked name="pdf_[% pdf.id %]" id="pdf_checkbox_[%
```

```
pdf.id %]" /></td>
  <td><a href="[% h.url_for('pdf_download', filename => pdf.name ) %]">[% pdf.name | htm
l %]</a></td>
  <td>[% pdf.page_count %]</td>
</tr>

[% END %]

</table>

</form>

[% END %]

@@ layouts/default.html.tt
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
  <head>
    <title>[% title %]</title>
  </head>

  <body>
    <h1>[% title %]</h1>

    [% IF h.flash('error') %]
      <p>!! <span id="pdf_merge_error">[% h.flash('error') %]</span> !!</p>
    [% END %]

    [% IF h.flash('message') %]
      <p>!! <span id="pdf_merge_message">[% h.flash('message') %]</span> !!</p>
    [% END %]

    [% content %]

    <div id="footer">
      &copy; 2012
    </div>

  </body>
</html>
```