I'm not robot
reCAPTCHA
Continue

Fpdf output multiple files

#1 / Dec 20, 2009 10:49pm Hi, I am using CI and FPDF with FPDFI, I have it set up and working pretty well, and have been using theses to generate a lot of PDF's in the last couple of years, but now I have a different issue, I am trying to generate a bunch of separate PDF's inside a for loop. The Basic's of my controller are;load->library('pdf'); // Load and initialise CI Email library $this->load->library('email'); $this->email->initialize(array('mailtype' => 'html')); // start loop - approx 15 records $monthlies = $this->invoice->getMonthlies(); if($monthlies) { foreach($monthlies as $monthly) { // Load CI/FPDF library $this->load->library('pdf'); // Create PDF (I have remove unnesscary code that pulls data for PDF.) // Make a random filename $filename = md5(time() * rand(100, 1000)) .'.pdf'; $this->load->view('pdf/invoice', array('filename' => $filename)); // validate customer email and sent PDF. if(valid_email($email)) { $this->email->from($this->config->item('admin_mail')); $this->email->to($email); $this->email->reply_to($this->config->item('admin_mail')); $this->email->subject('Invoice #'. $invoice_id); $this->email->message($this->load->view('admin/email', array('customer' => $customer, 'invoice' => $invoice), TRUE)); // Attach my PDF, send and clear email + all attachemnts $this->email->attach($filename); $this->email->send(); $this->email->clear(TRUE); } // end loop } } // finished redirect(); } } This script run's and emails each record as expected, but it is concatinating my PDF's in to one PDF, EG; 1st loop, correct single page PDF, 2nd loop, 1st page is the previous record, second is a blank PDF page ? 15th record, as the first PDF correct, and a random amount of blank pages. My PDF View generates a single PDF fine, I have tried to reset thte PDF buffer after my output;$this->pdf->Output($filename, 'F'); $this->pdf->closeParsers(); $this->pdf->Close(); unset($this->pdf->buffer); $this->pdf->buffer = ''; $this->pdf->buffer = NULL; My CI PDF library includes the fpdfi library like so;AddFont('HelveticaNeue-Light','','HelveticaNeue-Light.php'); $this->AddFont('HelveticaNeue-Light','B','HelveticaNeue-Bold.php'); } function Header() { $ci =& get_instance(); $this->SetY(40); } function Footer() { } } #2 / Dec 20, 2009 11:05pm I haven't played around with FPDF in a while, but I'm pretty certain you'll either need to delete the PDF file that's generated before creating a new one with the same name, or you'll need to use a different name for each PDF file created. #3 / Dec 21, 2009 12:48am I haven't played around with FPDF in a while, but I'm pretty certain you'll either need to delete the PDF file that's generated before creating a new one with the same name, or you'll need to use a different name for each PDF file created. Thanks Aken, Sorry that was an error in my post, In this instance, the filename is generated by the mySQL auto incrementing ID, and a rand md5() has in the file name, is is never the same. Here is the working code, CI is great for including a sinlge library instance, but there is no way (That I know of) to unset a library. load->library('email'); $this->email->initialize(array('mailtype' => 'html')); // start loop - approx 15 records $monthlies = $this->invoice->getMonthlies(); if($monthlies) { foreach($monthlies as $monthly) { // Load CI/FPDF library $this->load->library('pdf'); // Create PDF (I have remove unnesscary code that pulls data for PDF.
) // Make a random filename $filename = md5(time() * rand(100, 1000)) .'.pdf'; // generate a random varible name $pdf = time(); // assign my PDF Class to my random name. $$pdf = new Pdf(); // Pass in my filename for output AND my randomly named PDF class $this->load->view('pdf/invoice', array('pdf' => $$pdf, 'filename' => $filename)); // unset my random PDF name - Do I need to, unsure, but heck, why not ! $pdf = md5(time()); // validate customer email and sent PDF. if(valid_email($email)) { $this->email->from($this->config->item('admin_mail')); $this->email->to($email); $this->email->reply_to($this->config->item('admin_mail')); $this->email->subject('Invoice #'. $invoice_id); $this->email->message($this->load->view('admin/email', array('customer' => $customer, 'invoice' => $invoice), TRUE)); // Attach my PDF, send and clear email + all attachemnts $this->email->attach($filename); $this->email->send(); $this->email->clear(TRUE); } // end loop } } // finished redirect(); } } #5 / Jan 29, 2012 10:58pm Hi, I am using CI and FPDF with FPDFI, I have it set up and working pretty well, and have been using theses to generate a lot of PDF's in the last couple of years, hi i have problem setting up fpdf, can u please help me?? thanks in advance This tutorial provides an overview of the FPDF functionality and two examples using the PHP object-oriented approach to get you started with building your own PDFs. PHP allows you to generate PDF files dynamically, which can be useful for a variety of tasks. FPDF is a free PHP class containing a number of functions that let you create and manipulate PDFs. PDFlib The PHP API contains a number of functions for handling PDF files designed to be used with the PDFlib. Although extensive, this library is not free for commercial use. A free version called PDFlib Lite is available for personal use, but is limited in functionality. To use the full PDFlib library you must purchase a rather expensive license. Why FPDF? An alternative way of generating PDF files with PHP is using FPDF, a free PHP class containing a number of functions for creating and manipulating PDFs. The key word here is free.
You are free to download and use this class or customise it to fit your needs. In addition to being free, it's also simpler to use than PDFlib. The PDFlib needs to be installed as an extension in your PHP package, whereas FPDF can just be included in your PHP script and it's ready to use. Creating PDF files To get started, you will need to download the FPDF class from the FPDF Web site and include it in your PHP script like this: require('fpdf.php'); Below is an example of how you can generate a simple PDF using FPDF. We begin by creating a new FPDF object with: $pdf= new FPDF(); The FPDF constructor can take the following parameters: |>String orientation (P or L) — portrait or landscape |>String unit (pt,mm,cm and in) — measure unit |>Mixed format (A3, A4, A5, Letter and Legal) — format of pages Next, we are going to set some document properties: $pdf->SetAuthor('Lana Kovacevic'); $pdf->SetTitle('FPDF tutorial'); Because we want to use the same font throughout the whole document, we can set it before we create a page. $pdf->SetFont('Helvetica','B',20); $pdf->SetTextColor(50,60,100); The SetFont function takes three parameters; the font family, style and size. We are using Helvetica, Bold and 20 points, which will be applied to the title of our document. You can either use one of the regular font families or set up a different one using the AddFont () function. With SetTextColor () we are also setting the font colour for the entire document. The colours can be represented as RGB or grey scale.

Here we are using RGB values. Now that that's done, let's set up a page for our PDF document. $pdf->AddPage('P'); $pdf->SetDisplayMode(real,'default'); You can pass the AddPage () a parameter of "P" or "L" to specify the page orientation. I've used "P" for portrait. The SetDisplayMode function determines how the page will be displayed. You can pass it zoom and layout parameters. Here we're using 100 percent zoom and the viewer's default layout. Now, that we've set up a page, let's insert an image to make it look nicer and make it a link while we're at it. We'll display the FPDF logo by calling the Image function and passing it the following parameters — name of the file, the dimensions and the URL. $pdf->Image('logo.png',10,20,33,0,' ',' '); You could have also inserted the link with: $pdf->Link(10, 20, 33,33, ' '); Now let's make a title for our document with a border around it. $pdf->SetXY(50,20); $pdf->SetDrawColor(50,60,100); $pdf->Cell(100,10,'FPDF Tutorial',1,0,'C',0); The SetXY function sets the position of x and y coordinates, where we want the title to appear.
SetDrawColor will set the colour of the border, using RGB values. After that's done, we call the Cell function to print out a cell rectangle along with the text of our title. We are passing the function the following parameters; width, height, text, border, ln, align and fill. The border is either 0 for no border or 1 for frame. For ln we are using the default value 0, "C" to centre align the text inside it and 0 for fill. Had we used 1 for fill the rectangle would have been coloured in. With a value of 0 we are making it transparent. Now we want to write the main text to the PDF, that is display a little message. $pdf->SetXY(10,50); $pdf->SetFontSize(10); $pdf->Write(5,'Congratulations! You have generated a PDF. '); Again we are setting the x and y positions of the text, but this time we are reducing the font size with the SetFontSize function. The write function will print the text to a PDF. The parameter 5 will set the line height. This is only relevant however, if there are multiple lines of text. Finally we want to send the output to a given destination, using the Output function. $pdf->Output('example1.pdf','I'); Here we are passing the function the name of the file and the destination, in this case "I". The "I" parameter will send the output to the browser. Putting it all together: SetAuthor('Lana Kovacevic'); $pdf->SetTitle('FPDF tutorial'); //set font for the entire document $pdf->SetFont('Helvetica','B',20); $pdf->SetTextColor(50,60,100); //set up a page $pdf->AddPage('P'); $pdf->SetDisplayMode(real,'default'); //insert an image and make it a link $pdf->Image('logo.png',10,20,33,0,' ',' '); //display the title with a border around it $pdf->SetXY(50,20); $pdf->SetDrawColor(50,60,100); $pdf->Cell(100,10,'FPDF Tutorial',1,0,'C',0); //Set x and y position for the main text, reduce font size and write content $pdf->SetXY (10,50); $pdf->SetFontSize(10); $pdf->Write(5,'Congratulations! You have generated a PDF.'); //Output the document $pdf->Output('example1.pdf','I'); ?> Now that you've learnt how to generate a simple PDF, let's see what else we can do with FPDF. The example code below demonstrates how to make a header and a footer for your document. Image('logo.png',10,8,33); $this->SetFont('Helvetica','B',15); $this->SetXY(50, 10); $this->Cell(0,10,'This is a header',1,0,'C'); } function Footer() { $this->SetXY(100,-15); $this->SetFont('Helvetica','I',10); $this->Write (5, 'This is a footer'); } } $pdf=new PDF(); $pdf->AddPage(); $pdf->Output('example2.pdf','D'); ?> As you can see we are creating a child class of FPDF using inheritance and setting up the behaviour for both the Header and Footer functions. We then create a new object of this PDF class and add a page to our document. The AddPage () will automatically call the Header and Footer. Finally, we output it to a file called example2.pdf, this time using the "D" option for the sake of the example. This will send the file to the browser and open a dialog box, prompting the user to save the file. Now that you have an understanding of how FPDF works, go ahead and play with some of its functions and have fun building your own PDFs. For the full list of functions, refer to the FPDF Web site. Published: February 20, 2008, 3:15 AM PST Modified: October 6, 2011, 1:53 AM PDT See more Developer