

[Click Here](#)



Hello, I need to create a single page (continuous) PDF file. I'm trying to figure out how to get the measured height to use with setPaper method. This is what I'm trying to do: use Dompdf\Dompdf; \$dompdf = new Dompdf(); \$dompdf->loadHtml(\$html); \$dompdf->render(); \$frame = \$dompdf->getTree()->get_frame(0); \$height = \$frame->get_style()->height; The problem is that this \$height variable contains an string "auto". I need the exact measured size by the renderer. Thanks. Creating PDF documents is a common requirement in web applications, especially for generating invoices, receipts, certificates, tickets, and various reports. In this comprehensive tutorial, we'll delve into using Laravel and DomPDF to generate PDF documents with images and CSS. We'll cover configuration options, design considerations, output size, performance, and database queries. Additionally, we'll discuss tips and tricks for handling page breaks, loading images using base64, and more. Prerequisites Before we start, ensure you have the following installed: PHP >=8.2 Composer 2+ Laravel 10+ Introduction DomPDF is a popular PHP library that allows you to generate PDF documents from HTML content. It supports CSS styling, images, and various configuration options. By integrating DomPDF with Laravel, you can easily create sophisticated PDF documents using Blade templates and Laravel's powerful features. Other popular PDF libraries include TCPDF, FPDF, and Snappy. However, DomPDF is widely used due to its ease of integration and robust feature set. In this tutorial, we'll walk through the process of setting up a Laravel project, configuring DomPDF, creating a controller to handle PDF generation, designing a Blade template for the PDF content, adding routes, and optimizing performance. We'll also discuss advanced configuration options and provide tips and tricks for generating high-quality PDF documents. Assumptions This tutorial assumes you have a basic understanding of Laravel and PHP. If you're new to Laravel, consider going through the official Laravel documentation to familiarize yourself with the framework. Other wise you can follow the Laravel Bootcamp to get started with Laravel. Step 1: Setting Up Laravel Project First, create a new Laravel project if you don't already have one, or use an existing project. Of course, you can skip this step if you already have a Laravel project. composer create-project --prefer-dist laravel/laravel pdf-tutorial cd pdf-tutorial Next, install DomPDF: composer require barryvdh/laravel-dompdf Publish the configuration file: php artisan vendor:publish --provider="Barryvdh\DomPDF\ServiceProvider" Step 2: Configuring DomPDF Open the config/dompdf.php file. The configuration file contains various options for customizing the PDF output. Here you can set various options including the default paper size, orientation, font, and more. Paper size: You can set the default paper size. 'default_paper_size' => 'a4'. Orientation: Set the default orientation (portrait or landscape). 'orientation' => 'portrait'. Font: You can specify the default font and add custom fonts. 'default_font' => 'serif', Step 3: Creating a Controller Create a controller to handle PDF generation: php artisan make:controller PdfController In app/Http/Controllers/PdfController.php, add the following code: