











|  |  |  |  |
|--|--|--|--|
|  <b>niklasravnsborg</b> | Merge pull request #201 from PanovAlexey/main                        |  a5f5c22 on Jun 8, 2021 |  56 commits |
|  .github/ISSUE_TEMPLATE | Update issue templates   | 3 years ago  |  |
|  src                    | feat: Add the ability to use and customize files in PDF/A-1b format. | 16 months ago  |  |
|  tests                  | Use new dummy image provider   | 3 years ago  |  |
|  .gitignore             | init   | 7 years ago  |  |
|  README.md              | Update Readme  | 3 years ago  |  |
|  composer.json          | Update to mpdf 8   | 3 years ago  |  |
|  phpunit.xml            | Add some basic PDF tests   | 4 years ago  |  |

 README.md

# Laravel PDF: mPDF wrapper for Laravel 5

Easily generate PDF documents from HTML right inside of Laravel using this mPDF wrapper.

## Installation

Require this package in your `composer.json` or install it by running:

```
composer require niklasravnsborg/laravel-pdf
```

Note: This package supports auto-discovery features of Laravel 5.5+, You only need to manually add the service provider and alias if working on Laravel version lower then 5.5

To start using Laravel, add the Service Provider and the Facade to your `config/app.php` :

```
'providers' => [
    // ...
    niklasravnsborg\LaravelPdf\PdfServiceProvider::class
]

'aliases' => [
    // ...
    'PDF' => niklasravnsborg\LaravelPdf\Facades\Pdf::class
]
```

Now, you should publish package's config file to your config directory by using following command:

```
php artisan vendor:publish
```

## Basic Usage

To use Laravel PDF add something like this to one of your controllers. You can pass data to a view in `/resources/views`.

```
use PDF;

function generate_pdf() {
    $data = [
        'foo' => 'bar'
    ];
    $pdf = PDF::loadView('pdf.document', $data);
    return $pdf->stream('document.pdf');
}
```

## Other methods

It is also possible to use the following methods on the `pdf` object:

- `output()` : Outputs the PDF as a string.
- `save($filename)` : Save the PDF to a file
- `download($filename)` : Make the PDF downloadable by the user.
- `stream($filename)` : Return a response with the PDF to show in the browser.

## Config

If you have published config file, you can change the default settings in `config/pdf.php` file:

```
return [
    'format'           => 'A4', // See https://mpdf.github.io/paging/page-size-orientation.html
    'author'           => 'John Doe',
    'subject'           => 'This Document will explain the whole universe.',
    'keywords'          => 'PDF, Laravel, Package, Peace', // Separate values with comma
    'creator'           => 'Laravel Pdf',
    'display_mode'      => 'fullpage'
];
```

To override this configuration on a per-file basis use the fourth parameter of the initializing call like this:

```
PDF::loadView('pdf', $data, [], [
    'format' => 'A5-L'
])->save($pdfFilePath);
```

You can use a callback with the key 'instanceConfigurator' to access mpdf functions:

```
$config = ['instanceConfigurator' => function($mpdf) {
    $mpdf->SetImportUse();
    $mpdf->SetDocTemplate('/path/example.pdf', true);
}]

PDF::loadView('pdf', $data, [], $config)->save($pdfFilePath);
```

## Headers and Footers

If you want to have headers and footers that appear on every page, add them to your `<body>` tag like this:

```
<htmlpageheader name="page-header">
    Your Header Content
</htmlpageheader>

<htmlpagefooter name="page-footer">
    Your Footer Content
</htmlpagefooter>
```

Now you just need to define them with the name attribute in your CSS:

```
@page {
    header: page-header;
    footer: page-footer;
}
```

Inside of headers and footers `{PAGENO}` can be used to display the page number.

## Included Fonts

By default you can use all the fonts [shipped with mPDF](#).

## Custom Fonts

You can use your own fonts in the generated PDFs. The TTF files have to be located in one folder, e.g. `/resources/fonts/`. Add this to your configuration file (`/config/pdf.php`):

```
return [
    // ...
    'font_path' => base_path('resources/fonts/'),
    'font_data' => [
        'examplefont' => [
            'R' => 'ExampleFont-Regular.ttf', // regular font
            'B' => 'ExampleFont-Bold.ttf', // optional: bold font
            'I' => 'ExampleFont-Italic.ttf', // optional: italic font
            'BI' => 'ExampleFont-Bold-Italic.ttf' // optional: bold-italic font
            // 'useOTL' => 0xFF, // required for complicated langs like Persian, Arabic and Chi
            // 'useKashida' => 75, // required for complicated langs like Persian, Arabic and Chi
        ]
        // ...add as many as you want.
    ]
    // ...
];
```

Note: If you are using `laravel-pdf` for producing PDF documents in a complicated language (like Persian, Arabic or Chinese) you should have `useOTL` and `useKashida` indexes in your custom font definition array. If you do not use these indexes, your characters will be shown dispatched and incorrectly in the produced PDF.

Now you can use the font in CSS:

```
body {
    font-family: 'examplefont', sans-serif;
}
```

## Set Protection

To set protection, you just call the `SetProtection()` method and pass an array with permissions, an user password and an owner password.

The passwords are optional.

There are a fews permissions: 'copy', 'print', 'modify', 'annot-forms', 'fill-forms', 'extract', 'assemble', 'print-highres'.

```
use PDF;

function generate_pdf() {
    $data = [
        'foo' => 'bar'
    ];
    $pdf = PDF::loadView('pdf.document', $data);
    $pdf->SetProtection(['copy', 'print'], '', 'pass');
    return $pdf->stream('document.pdf');
}
```

Find more information to `SetProtection()` here: <https://mpdf.github.io/reference/mpdf-functions/setprotection.html>

## Testing

To use the testing suite, you need some extensions and binaries for your local PHP. On macOS, you can install them like this:

```
brew install imagemagick ghostscript
pecl install imagick
```

## License

Laravel PDF is open-sourced software licensed under the [MIT license](#)


## About


👀 Looking for new maintainers, please reach out by opening an issue.


laravel


mpdf

pdf-generation

 Readme

 643 stars

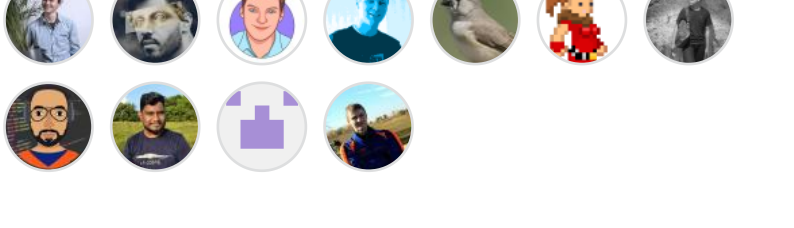
 22 watching

 175 forks

## Releases

🏷 21 tags

## Contributors



## Languages

