

# Creating PDFs

## ON THIS PAGE

- Sizing the pdf
- Using a predefined format
- Setting margins
- Headers and footers
- Backgrounds
- Tagged (accessible) pdf
- Landscape orientation
- Scale
- Only export specific pages
- Set the initial page number
- Output directly to the browser

Browsershot will save a pdf if the path passed to the `save` method has a `pdf` extension.

```
// a pdf will be saved
Browsershot::url('https://example.com')->save('example.pdf');
```

Alternatively you can explicitly use the `savePdf` method:

```
Browsershot::url('https://example.com')->savePdf('example.pdf');
```

You can also pass some html which will be converted to a pdf.

```
Browsershot::html($someHtml)->savePdf('example.pdf');
```

If you need the base64 version of a PDF you can use the `base64pdf` method. This can come in handy when you don't want to save the screenshot on disk in environments like Heroku that don't allow you to save a file. You can then proceed to create the file and upload it directly as a base64 string using a package like Laravel Media Library.

```
$base64Data = Browsershot::url('https://example.com')
```

```
->base64pdf());
```

## Sizing the pdf

You can specify the width and the height.

```
Browsershot::html($someHtml)
  ->paperSize($width, $height)
  ->save('example.pdf');
```

Optionally you can give a custom unit to the `paperSize` as the third parameter.

## Using a predefined format

You can use the `format` method and provide a format size:

```
Browsershot::url('https://example.com')->format('A4')->save('example.pdf');
```

The `format` options available by puppeteer are:

```
Letter: 8.5in x 11in
Legal: 8.5in x 14in
Tabloid: 11in x 17in
Ledger: 17in x 11in
A0: 33.1in x 46.8in
A1: 23.4in x 33.1in
A2: 16.54in x 23.4in
A3: 11.7in x 16.54in
A4: 8.27in x 11.7in
A5: 5.83in x 8.27in
A6: 4.13in x 5.83in
```

## Setting margins

Margins can be set.

```
Browsershot::html($someHtml)
  ->margins($top, $right, $bottom, $left)
  ->save('example.pdf');
```

Optionally you can give a custom unit to the `margins` as the fifth parameter.

# Headers and footers

By default a PDF will not show the header and a footer generated by Chrome. Here's how you can make the header and footer appear. You can also provide a custom HTML template for the header and footer.

```
Browsershot::html($someHtml)
->showBrowserHeaderAndFooter()
->headerHtml($someHtml)
->footerHtml($someHtml)
->save('example.pdf');
```

In the header and footer HTML, any tags with the following classes will have its printing value injected into its contents.

- `date` formatted print date
- `title` document title
- `url` document location
- `pageNumber` current page number
- `totalPages` total pages in the document

To hide the header or footer, you can call either `hideHeader` or `hideFooter`.

## Backgrounds

By default, the resulting PDF will not show the background of the html page. If you do want the background to be included you can call `showBackground`:

```
Browsershot::html($someHtml)
->showBackground()
->save('example.pdf');
```

If you want a transparent background color instead of the default white, you can call `transparentBackground`:

```
Browsershot::html($someHtml)
->transparentBackground()
->save('example.pdf');
```

## Tagged (accessible) pdf

Call `taggedPdf` if you want the resulting pdf to be tagged (accessible).

```
Browsershot::html($someHtml)
->taggedPdf()
```

```
->save('example.pdf');
```

## Landscape orientation

Call `landscape` if you want to resulting pdf to be landscape oriented.

```
Browsershot::html($someHtml)
->landscape()
->save('example.pdf');
```

## Scale

Scale can be set. Defaults to 1. Scale amount must be between 0.1 and 2.

```
Browsershot::html($someHtml)
->scale(0.5)
->save('example.pdf');
```

## Only export specific pages

You can control which pages should be export by passing a print range to the `pages` method. Here are some examples of valid print ranges: 1, 1-3, 1-5, 8, 11-13.

```
Browsershot::html($someHtml)
->pages('1-5, 8, 11-13')
->save('example.pdf');
```

## Set the initial page number

You can set the initial page number with `initialPageNumber`. Here's an example that shows how to set the first page to number 8.

```
Browsershot::html($someHtml)
->showBrowserHeaderAndFooter()
->hideFooter()
->headerHtml('<span class="pageNumber"></span>')
->initialPageNumber(8)
->save('example.pdf');
```

# Output directly to the browser


You can output the PDF directly to the browser using the `pdf()` method.

```
$pdf = Browsershot::url('https://example.com')->pdf()
```

[← Creating images](#)

[Creating HTML >](#)

 [Help us improve this page](#)

- [Vacancies](#)
- [About](#)
- [Blog](#)
- [Docs](#)**
- [Guidelines](#)
- [Log in](#) 

Kruikstraat 22, Box 12  
2018 Antwerp, Belgium  
[info@spatie.be](mailto:info@spatie.be)  
+32 3 292 56 79