Übersicht

Niirrty.Md2Pdf is a tool that generates a PDF File from one or more markdown files.

Installation

Niirrty.Md2Pdf comes as a PHAR archive and can be called if a usable PHP7.3+ is known at your system.

Simple unpack `md2pdf.zip` to a place of your choice.

If you are done with it you have to make it globally usable…

For unixoid systems

As symbolic link (symlink)

Create a symbolic link from md2pdf.phar to a folder that is a part of your PATH environment variable.

for example

```
sudo ln -s ~/programs/md2pdf/md2pdf.phar /usr/bin/md2pdf
```

As copy

You can also copy it to a folder that is a part of your PATH environment variable.

```
sudo cp ~/programs/md2pdf/md2pdf.phar /usr/bin/md2pdf
```

Make it executable

```
sudo chmod +x /usr/bin/md2pdf
```

If you now call it from somewhere it should work and show the version of used md2pdf

```
md2pdf --version
```

For Windows systems

Add the current location to PATH environment variable

Configuration

There are 2 different ways to configure md2pdf.

Interactive Configuration

Call

```
md2pdf init
```

inside the folder that contains the MD files.

Now you have to input the required data:

- **Init with defaults?** If you chose 'y' (yes) the default files for header template, user and logo image is enabled
- **Output PDF file** Here you have to insert the path of the resulting output PDF file.
- **Scan recursive for MD files?** Chose y if the current working dir should be scanned recursive for all available markdown files. In this case all found markdown files from sub folders are included in resulting documentation
- **Markdown file (optionally)** If you want only include one or more specific markdown files define it here. If not, or if you are done leave input empty and press `ENTER`.
- **Document title** This is the title of the resulting PDF document, not visible inside the PDF document.
- **Page title** The title text, eg. visible at each page header, if used.
- **Document header template file** (only if "init with defaults" is not chosed) The path of the HTML template that defines the page header.
- **Document footer template file** The path of the HTML template that defines the page footer.
- **User CSS files** (only if "init with defaults" is not chosed) The path of the CSS files that defines user styles.
- **TOC max level** If a automatic TOC (table of contents) should be included here you have to define the max level of headings (Level 1 = h1 - Level 6 = h6) If level is 0, no TOC is generated.

If all these steps are done all is saved inside `md2pdf.json` into current working directory

If "init with defaults" is used, the directory `.md2pdf` is created and the header template, user style, logo image and initial required fonts are inserted. If you want to use other fonts copy them to defined `.md2pdf/fonts` folder and use them inside the CSS stylesheet file `.md2pdf/md2pdf.css`

If you want to write configuration data to a different config json file you can define the file name by

```
md2pdf init --config-file=my-md2pdf-config.json
```

Config file

The config of a md2pdf project must be defined by file `md2pdf.json` inside the folder (no sub folder) where md2pdf is called. Or if another file name should be used it must be passed by `--config-file=CONFIG-FILE` option.

Example file

```
{
   "outputFile": "md2pdf-documentation.pdf",
   "recursive": false,
   "files": [],
   "header": ".md2pdf/header.tpl",
   "footer": ".md2pdf/footer.tpl",
   "userCSS": [ ".md2pdf/md2pdf.css" ],
   "documentTitle": "Md2Pdf Documentation",
   "pageTitle": "Md2Pdf Documentation",
   "tocMaxLevel": 3,
   "pageMarginTopIfHeader": 25
}
```

- **outputFile** `string` Here you have to insert the path of the resulting output PDF file.
- **recursive** `boolean` If enabled the current working dir should be scanned recursive for all available markdown files. In this case all found markdown files from sub folders are included in resulting documentation
- **files** `string[]` If no files are defined the current working dir is scanned for all available markdown files, depending to the `recursive` setting. If defined, only the files are scanned, defined here.
- **header** `string` The path of the HTML template that defines the page header.
- **footer** `string` The path of the HTML template that defines the page footer.
- **userCSS** `string[]` The paths of the CSS files that defines user styles.
- **documentTitle** `string` This is the title of the resulting PDF document, not visible inside the PDF document.
- **pageTitle** `string` The title text, eg. visible at each page header, if used.
- **tocMaxLevel** `integer` If a automatic TOC (table of contents) should be included here you have to define the max level of headings (Level 1 = h1 - Level 6 = h6) If level is 0, no TOC is generated.
- **pageMarginTopIfHeader** `integer` The top margin (mm) of page content if a page header is used (default=25)

The header template

If you want add a page header, show on each PDF page (like a logo and/or text) you have to define a HTML template.

Here is a short example:

```
<table class="header">
   <tr>
      <td class="logo">
         <img src="file://{CWD}/.md2pdf/logo.png" style="width: 40mm; height: auto; border: none;">
      </td>
      <td class="title">
         {TITLE}
      </td>
      <td class="pageInfo">
         Seite 4
      </td>
   </tr>
</table>
```

The used replacements are the only usable replacements:

- `{CWD}`: The current working directory where md2pdf is called
- `{TITLE}`: The page (header) title, defined by `md2pdf.json` config file or generated by calling `md2pdf init`
- `4` : The current page number

Images should be declared by file:// scheme!

It should be styled by a user CSS file.

The footer template

Similar to header template…

User CSS stylesheet

You can define CSS styles for output formatting.

If you want to see the HTML structure, styled with the CSS you can call the:

```
md2pdf build -d
```

command. The `-d` or `--dump-html` option do the same like without the option but also dumps the HTML to STDOUT that is converted to PDF.

So you can easy see the HTML that you can style with CSS.

Fonts

There are no fonts configurable. You can use basic serif, sans-serif and monospace fonts.

Build the PDF output

Finally you should call

```
md2pdf build
```

to generate/build the PDF output file.

If you want more output info you can use the `--verbose` or `-v` option

```
md2pdf build --verbose
```

If you want to read configuration data from a different config json file you can define the file name by

```
md2pdf build --config-file=my-md2pdf-config.json
```

Supported markdown

The commonmark spec is provided widely.

Syntax highlighting

Also some specific code, marked with a language is shown syntax highlighted.

Images

Images supports 3 optional attributes

```
![Alt text](path/to/image.png|width=150|height=auto|class=left)
```

- `width`: A valid image width declaration
- `height`: A valid image height declaration
- `class`: The name of a associated CSS class

There a 2 different predefined css classes.

- `left`: floats the image to left side
- `right`: floats the image to right side

Pagebreak

Manual page breaks are supported by

```
!!pagebreak!!
```