MarkDown Thesis FrameWork

Jan Sobeslav

February 2018

Abstrakt

Lorem Ipsum dolor sit amet

Obsah

Seznam tabulek

Seznam obrázků

MarkDownThesisFW

Overview and philoshophy

As stated in the readme, this framework is a tool that utilizes Pandoc to create beautiful standalone ebooks and documents from multiple MarkDown markup files. The text contents are written in plain text, and can therefore be tracked by version control system, like Git, and before exporting to pdf or epub, the MD code is translated to, and interpreted as LaTeX.

The main reasons, why I decided to try this approach, instead of continuing to use any office document processing tool, were following:

- Modular nature: I wanted to divide a lengthy document into more managable set of files - chapters. That way, I can see two chapters side by side and modify them parallelly without scrolling all the

time (I figured out, that such feature would be invaluable, only after writing a 13k words bachelor thesis)
- Version control: I thought it would be useful to see different versions of the document, and changes made between them. Moreover, remote VCS makes perfect backups, and even makes possible collaboration way easier
- Clearer style: Since MD markup is quite limited, I like to think of it as of a benefit: I hope it will give the document more unified, more readable look

Features checklist

There are certain features this framework in general, and other tools I use, must provide for me. Some features must be satisfied by recommended text processing program, other by the MD-to-PDF compilation tool. It's important to note, that theese points are my personal requirements, and I present them only to give an idea on the philosophy behind the project, and what could it give you.

As noted further in the docs, the framework relies on Pandoc tool, and I personally chosed Haroopad as text processor, which suits me the most.

My text processor requirements:

- [x] Live compilation result preview
  - I need to see the result of MD processing as I write. I prefer two-pane layout, instead of WYSIWIG, and this requirement is satisfied by Haroopad
- [x] Ability to view multiple files simoutanously
  - Haroopad works in windowed mode, and therefore it is possible to arrange the files for convenience
- [x] Multiple monitor support
  - Again, windowed mode enables spreading the work on multiple monitors. This is feature, which might not be able to accomplish with a single-window IDE (which could, however, at least support viewing multiple files at the same time)
- [ ] Tabbed interface of opened documents
  - Haroopad does not have this feature. Instead, it works in windowed mode, which is not perfect to me personally, but it is satisfactory enough
- [x] Diacritics support (UTF8)
  - Haroopad works well with UTF-8

Framework requirements and goals:

- [x] Git support
  - MD files are basically plain-text and therefore perfectly compatible with any VCS
- [x] Export: Export to PDF (EPUB), merge of multiple files
  - Pandoc does this, and much more
- [x] Content: images, code preview
  - Works like a charm, as seen in example chapter
- [x] Content: tables
  - It's difficult to maintain the tables in plain text, but it's possible, as seen in example chapter
- [x] Organization: Table of contents, list of tables, list of figures
  - Another few of nice features Pandoc provides
- [ ] Organization: Bibliography (References)
  - Hasn't been tested yet, but there is Pandoc support for this, too
- [ ] Organization: pagebreaks, heading numbering
  - Hasn't been tested yet
- [x] Style: Template support
  - It's not perfect enough in current version of the framework, as described in first News entry, but Pandoc does have the template support
- [ ] Custom formatting / style
  - Hasn't been tested properly

- [ ] Formalized FW
  - In the end, I want to present a publishable version of the project, with accurate docs and all features working and tested

News

> I have just finished the first working verion of the framework, and will test it on a real example soon. When that happens, I will probably update the tool with some fixes, or additional options. I am quite happy with the results so far, with the only exception being the template format. I wanted the template to be written in MD as well and to be translated to LaTeX during the compillation, but there were errors, and that's why I left the template in LaTeX. I will try to look into it later. - February 2018

Docs

Dependencies

The framework requires Pandoc tool installed. If You need to export to PDF, you will also need MiKTeX, as Pandoc docs recommend.

I also had an issue with following error: `pdfTeX error (font expansion): auto expansion is only possible with scalable fonts`. I found solution in this latextemplates issue thread, where user by the name koppor recommended installing `cm-super` MiKTeX plugin, which has worked for me.

MarkDown processing

I use Haroopad to process MD files, since I like the two-pane interface more, than any WYSIWIG, like what Typora offers. Unfortunatelly, Haroopad has not been updated for a long time, and might have some issues, especially in the future. Nevertheless it still works for me, which is why I am leaving it here as a recommendation.

A decent list of other recommendations is available in this SitePoint article.

Moreover: You can use any IDE with MD highlight support: NetBeans, for example, offer optional MarkDown plugin (with live preview functionality), and enables user to split the window area, so that only issue I have had with it, was that the IDE does not break lines, which could be fixed in a few ways (with `Toggle line wrap` plugin, for example).

Directory structure:

- `_script`: Operating system batch command files that compile chapters and template into an output in requested format
  - `windows_cmd.bat`: Windows batch file; tested on Windows 10. Check the script variables section bellow
  - Other OS batch command files are missing; If you are writing one on your own, please see the `windows_cmd.bat` comments, and, please, contribute to the project with your solution
- `_temp`: Temporary files required for compillation process
- `_output`: Folder, where the output file will be saved
- `metadata`: The document meta information declarations, containing data like author name, or list of sources
  - `metadata.yaml`: Book meta data, a list of data like author name, in YAML format
  - `bibliography.bib`: Bibliography, a list of references in BibTeX format
- `template`: The general output document markup, style and meta data

  - `template.tex`: A template file, containing intro, tables of contents, figures, and tables, and other document parts, in Pandoc LaTeX template format You can replace it with any user contributed templates
  - `style_metadata.yaml`: Style-related meta data, containing data like paper size, or page margins
  - `citation_style.csl`: A citation style declaration. Default file is IEEE standard, but feel free to replace it with any other. You can find thousands of CSL files on Zotero Style Repository
  - `epub_style.css`: A stylesheet declaration for epub format
- `_script`: Operating system batch command files that compile chapters and template into an output in requested format
  - `windows_cmd.bat`: Windows batch file; tested on Windows 10. Check the script variables section bellow
  - Other OS batch command files are missing; If you are writing one on your own, please see the `windows_cmd.bat` comments, and, please, contribute to the project with your solution
- `_temp`: Temporary files required for compillation process
- `_output`: Folder, where the output file will be saved

Use

First of all modify all the content files, and the meta data as well. Once You want to compile the results, navigate to the `_script` folder and run whichever You need (either through CLI, or GUI, if your OS allows to run the script from there)

You might want to look into the script, and modify following variables:

- `extension` (default: `pdf`): Result output format; I will test the results for `pdf` and `epub` only, but other values are possible, as seen in Pandoc docs
- `result` (default: `result`): Result output filename in `_output`
- `template` (default: `template.tex`): A chosen LaTeX template file from `template` folder

Again, the script is built on Pandoc. You should check their docs as well.

Functions example

Chapter One

Nulla ligula velit, nec ornare felis placerat sed. Morbi accumsan, ligula commodo varius viverra, lectus sem interdum sapien, eget hendrerit velit eros sit amet mi. Fusce convallis est pulvinar, sollicitudin sem eget, suscipit turpis. Vivamus euismod fringilla mauris, vel porta risus porta quis. Praesent ultrices auctor urna, ut scelerisque arcu euismod at. Vivamus odio elit, tempor quis pellentesque at, sodales non dolor. Suspendisse et lobortis urna. Vestibulum ullamcorper purus nibh, vitae dapibus nisl fermentum et.

> Etiam sagittis ut nisl ut cursus. Fusce id dapibus lorem, in venenatis nibh. Mauris interdum congue dolor sit amet mollis. Morbi pellentesque tellus et libero pulvinar - Author

Sed sed scelerisque nibh. Etiam nec iaculis nisl, quis fringilla metus. Maecenas vel justo id lectus consequat scelerisque. Follows section break.

Morbi nec turpis placerat, consectetur sapien nec, consectetur leo. Pellentesque ac neque molestie, consectetur sapien quis, cursus erat. Nam elementum neque non nisi ullamcorper vulputate.

Table	Example
Github	Style
10	45

Aenean congue risus ante, in malesuada elit sagittis et. Pellentesque ex ex, viverra quis nisl at, tempor venenatis mi. Aliquam erat volutpat. Nunc quis dictum lectus. Nam lacinia pellentesque elit.

Chapter Two

Lorem ipsum dolor sit amet, changed adipiscing elit. Vestibulum feugiat viverra ligula, non condimentum purus elementum egestas. Pellentesque ultrices, nisl id consequat ultricies, felis orci dictum metus, eget congue urna libero non dui. Vivamus non nisi eu turpis mollis tempus. Aliquam maximus ullamcorper ante sit amet accumsan. Nam consequat, eros nec feugiat scelerisque, lorem sapien vehicula nulla, in fermentum diam orci vitae nulla. Praesent at nisl ut nulla sagittis mollis. Etiam arcu elit, sagittis ac fringilla in, consequat in sapien. Fusce tincidunt dignissim facilisis. Follows page break.

Nulla lobortis sem non varius pretium. Etiam luctus urna eget lorem posuere, auctor gravida mauris imperdiet. Praesent in porta nibh, eu feugiat mi. Maecenas et risus risus. Nullam eleifend elit a massa semper facilisis. Maecenas ultrices nulla pellentesque sem malesuada, a vehicula purus dapibus.

Vivamus dolor nibh, consequat non nisl vitae, sagittis tincidunt erat. Maecenas convallis, risus vel cursus varius, diam tellus tristique ex, elementum mattis ante enim in ante.

Obrázek 1: A magnificent animal

Donec euismod massa a magna blandit convallis. Ut vulputate risus augue, ut dictum diam feugiat non. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

Curabitur congue interdum ex a ornare. Some erat volutpat. In nec words orci, ac dapibus eros. Sed mauris mi, ullamcorper eget egestas eu, suscipit vel libero. In augue sapien, ultrices in congue vitae, commodo a purus. Donec ultrices ullamcorper mi at placerat. Morbi in gravida erat, nec laoreet leo. Aliquam semper ante vitae mi aliquam luctus. Vivamus dignissim sapien sapien, vel pellentesque nibh suscipit vel. Pellentesque enim turpis, tincidunt vel nisl nec, consequat molestie arcu. Donec porta risus risus, vestibulum vestibulum

ipsum iaculis vel. Curabitur a dapibus nibh. Aliquam sed vehicula ante. Etiam pharetra tellus quis ex eleifend ultricies.[1]

Aliquam erat volutpat. Fusce in thesis ultrices tortor, et ornare ligula pharetra a. Sed tristique diam et urna feugiat blandit. Cras arcu nulla, rhoncus sit amet aliquet nec, aliquam vitae risus. Etiam bibendum, ante quis hendrerit porttitor, orci elit tempor tortor, suscipit finibus mi `literal text`. Vestibulum et semper velit. Cras ligula sapien, sagittis id maximus ut, porta vitae orci. Duis ac leo libero.

```
main() {
    printf("hello, world");
}
```

Aliquam feugiat tempor vestibulum.

[1] A quick footnote example Etiam luctus urna