Vim as a Markdown Editor

Amolith

2020-04-30T23:06:59-04:00

Contents

I've recently decided to attempt to keep all of my notes and everything I've learned in a Zettelkasten. After reading Daryl Sun's blog post, I started looking more into the method and found it *incredibly* intriguing. I've tried the "Evernote way" of throwing everything I come across in a single place but it inevitable gets lost. I don't remember what it was called but I tried another app that actually tags your files and organises them in a nice manner. This worked well for the most part but the graphical client was badly optimised Electron and *very* heavy. I've also tried keeping notes in books but I was never really able to keep up with any of it. The thing that is especially compelling about a Zettelkasten is that I put *everything* I learn in a single text file but link around to as many different ideas as I can, drawing my *own* connections for me to rediscover later on.

Because it's all in a simple text file, I'm also able to create a keybinding in Sway that will open it in Vim, jump to the bottom, and have a nice markdown environment ready for me to write in. It did take a bit of configuration and looking around for different plugins but I'm very happy with what I have so far.

The first thing is telling Vim to treat all `.md` files as Markdown

```
" Treat all .md files as markdown
autocmd BufNewFile,BufRead *.md set filetype=markdown
```

Visuals

In a long text file with a great many lines, it can be useful to find your cursor quickly without having to search around the screen for it.

```
" Highlight the line the cursor is on
autocmd FileType markdown set cursorline
```

It can also be nice to not see a ton of [links](https://example.com) and **bold** or *italic* text everywhere. Sure, my eye has gotten used to it but still. I'd rather have my terminal actually render bold text as bold.

```
" Hide and format markdown elements like **bold**
autocmd FileType markdown set conceallevel=2
```

If you use the `vim-markdown` plugin mentioned further on, I recommend using its option for concealing rather than Vim's.

Spell check

One of the things every good editor needs is spell check and Vim is no exception. This line enables spell check with British English for all markdown files.

```
" Set spell check to British English
autocmd FileType markdown setlocal spell spelllang=en_gb
```

Here's a short crash course in Vim spelling commands: - `[s` to search for misspelled words above the cursor - `]s` to search for misspelled words below the cursor - `z=` to see replacement suggestions - `zg` to add the word to your dictionary

Goyo

The very first component is something I use across *all* markdown files. Goyo is one of the first plugins I install on any machine I'll be writing with. It enables a "distraction-free writing environment" and I absolutely love it. It disables pretty much all visual elements in Vim except for what mode you're in: visual, command, insert, etc. I have a keybinding set to quickly open/close Goyo because there is an odd issue when I switch workspaces to and away from Vim. With two taps of `Ctrl+g`, it's back to normal.

```
nnoremap <C-g> :Goyo<CR>
```

Another line in my Vim config automatically opens Goyo for all markdown files:

```
autocmd FileType markdown Goyo
```

vim-markdown

That latest plugin I installed is vim-markdown and it is *wonderful*. I really recommend reading about all of the options but here's what I have set.

```
" Configuration for vim-markdown
let g:vim_markdown_conceal = 2
let g:vim_markdown_conceal_code_blocks = 0
let g:vim_markdown_math = 1
let g:vim_markdown_toml_frontmatter = 1
let g:vim_markdown_frontmatter = 1
let g:vim_markdown_strikethrough = 1
let g:vim_markdown_autowrite = 1
let g:vim_markdown_edit_url_in = 'tab'
let g:vim_markdown_follow_anchor = 1
```

In addition to the rest of the awesome features, the main one I wanted is the last: `follow_anchor`. With this, I can create internal links within the same markdown document and jump between them with `ge`. It also lets me open both files and URLs from within Vim and without ever having to reach for the mouse.

General Vim things

Other, more general Vim settings that I use globally but might also be nice for editing markdown

```
" Have lines wrap instead of continue off-screen
set linebreak

" Gives Vim access to a broader range of colours
set termguicolors

" Converts tabs to spaces
set expandtab

" Use two spaces instead of tabs
set tabstop=2

" The same but for indents
set shiftwidth=2

" Keep cursor in approximately the middle of the screen
set scrolloff=12

" Disable mouse support
set mouse=
```

In all, I'm hoping that the work I've done today for improving my markdown workflow will help me create a more effective Zettelkasten. The *big* thing was really being able to follow internal links around because that's the main thing with keeping a Zettelkasten: following your ideas to see where they lead and discovering what connections you can make to form entirely new ideas. Mine will be stored in Gitea for now but I'm thinking about putting it here at some point. It would be cool to have a map of my own mind very easily accessible from anywhere.

Figure 1: screenshot of my setup

Edit

Time stamps

```
" Insert timestamp at the end of the line in this format:
    20200527T113245
nnoremap <C-t><C-s> m'A<C-R>=strftime('%Y%m%dT%H%M%S')<CR>
```

Portable `autocmds`

Put all the `autocmd` lines in the `if` statement so they don't throw errors when the config is added to a version of vim without `autocmd` support

```
" Only enable autocommands when Vim supports them
if has("autocmd")
    ""
    " Markdown Configuration
```

```
    ""
    " Spellcheck in British English
    autocmd FileType markdown setlocal spell spelllang=en_gb
    " Automatically open Goyo
    autocmd FileType markdown Goyo
    " Hide plaintext formatting and use color instead
    autocmd FileType markdown set conceallevel=3
    " Disable cursor line and column highlight
    autocmd FileType markdown set nocursorline
    autocmd FileType markdown set nocursorcolumn
endif
```

I won't keep editing this post to provide updates on my config. Instead, I recommend looking at my "production" version on Gitea.

This was posted as part of #100DaysToOffload, an awesome idea from Kev Quirk. If you want to participate, just write something every day for 100 days and post a link on social media with the hashtag!