

Website Setup

Website Setup

This post documents how my website is setup. Since I deploy it via github pages, that really amounts to how it is built.

The premise is that I have several component projects that all dump their build artifacts into a common root directory. I then run the a templater on that directory to produce the final website.

Tools

You will need the following special tools in your `PATH`.

Cobalt

cobalt.rs is a static site generator. I am currently using:

`Cobalt 0.7.2`

The best way to install cobalt is probably:

```
cargo install cobalt-bin
```

pandoc

pandoc is best described on their homepage:

If you need to convert files from one markup format into another,
pandoc is your swiss-army knife.

The version of pandoc I am using is:

`pandoc 1.19.2.1`

`Compiled with pandoc-types 1.17.0.5, texmath 0.9, skylighting 0.1.1.4`

I installed pandoc using brew:

```
brew install pandoc
```

blockpass

blockpass is a small utility I wrote. Top of tree should be fine for now. You can install by downloading and building it:

```
git clone https://github.com/ThermalSpan/blockpass.git
cd blockpass
cargo install
```

Dependences

These are the projects that contribute content to the website.

KaTeX

blockpass is a javascript library for math rendering. From Khan Academy! I am currently using the following tag:

v0.7.1

Pandoc supports KaTeX in html, so that's how I use it.

pandoc-posts

pandoc-posts is the repository where most of the posts live. Top of tree on the master branch should be sufficient. Drafts live in other branches.

thermalspan.github.io-source

thermalspan.github.io-source is kitchen sink for this website. All the templates, misc pages, and such live here. And for now the “build system” does too, but I'd like to rip that out.

Notes

The pandoc posts project is really important, and is where most of the content lives. I would like to keep it as a separate repository since it represents a good content divide. It also allows the issue tracker to be used solely for content related issues. I can also use the branches to cleanly mark drafts and such.

My short term goals are:

- Have three articles that aren't totally embarrassing in the pandoc-posts project
- Write a better build script / or find an off the shelf solution that meets my needs.
- Fixup the connect page

My medium term goals are:

- Template the posts to include download link to pdf version
- Include some projects
- Perhaps host some documentation, though this is pending having a project that's worth documenting
- To go with above, allow build script to build and include documentation in a clean way

My long term goals are:

- Seperate build system / templater from the webpage source
- Include fancier front page options for sorting through options
- Include a source option
- Include at least one webGL / javascript interactive project

Some nifty ideas that might be worth implementing:

- That progress bar that quantamagazine has on their articles is awesome.

Some Post ideas:

- Talk about cargo cache commands and how you use them
- Talk about nvim setup, or at least one useful trick for configuring SpaceVim
- Create Context Free Grammer cheatsheet
- Walk through a cool math problem to show off all that awesome KaTeX
- Introduce some notation and explain its tradeoffs. Like bra-ket, or Einstein summation.
- Take some notes on the many papers you think you've read, maybe a prolog implementation paper
- To maintain a list of other notable places on the web. Like other personal websites and such. Though, this might be better as a pure website for interactive purposes