Static Site Generator Daino: Landing page

The Example Homepage produced by daino list differences to other SSG and gives rationale for its design. It points to the source and how to use it to run your own web site.

1 Daino: A Static Site Generator

A static site generator designed by an academic to allow:

- web pages written as (Pandoc) markdown (with YAML header for title and `bibtex` references, etc.),
- page layout inspired by Tufte and using `w3c` framework to adapt to different screen sizes,
- publication list for download produced from `bibtex` database,
- offer printable `pdf` files for all content; for some directories compilation from multiple pages on a single *booklet* page,
- web site using multiple languages, with support for common shortcuts for typing (e.g. ae is changed to ä when acceptable)
- content and appearances (theme) separated,
- a single `yaml` file for setup, and
- a self-contained result which can be hosted on any web server.

The program runs on Debian Linux[1] locally on PC or Raspery Pi 4. Restricted accsss on some directories[2].

1.1 Software reuse:

Daino uses `pandoc` and other packages on `Hackage` (e.g. shake, twitch, scotty)[3]. Relies on `git` for version management and Debian Linux.

[1] Could run on Ubuntu or likely Windows(probably some adaptations needed).

[2] My provider uses `cpanel` and allows password protection to any directory.

[3] It was influenced by Chris Penner's slick, newer, and seemingly simpler is `Ema` by Sridhar Ratnakumar, but the documentation did not detail its features neither how it is built.

1.2 Example site

The example site shown here contains more information how to build a site with `daino`.

`Daino` can be installed from hackage or downloaded or cloned from `git clone https://github.com/andrewufrank/daino.git` and installed with `cabal install` or `stack install`[4].

The example site can be downloaded or cloned from `github` with `git clone https://github.com/andrewufr` In the `settings3.yaml` file replace `/home/frank/dainoTest_1_5_2` three times with the path to the directory which contains the test site.

To run the test site, start daino in it with `daino -qs` and render the result in a browser by opening `localhost:3000`. The web pages written in markdown can be edited and the server restarted to update the site.[5]

1.3 Running your own site

Copying the the test site to a suitable directory and edit the `settinsNN.yaml`[6] file found there is enough to start your own site with running `daino -qs` in this directory.

Delete the `.git` directory in the copy and restart git with `git init` to obtain version management for your site.

2 More information:

The ReadMe in the test site explain the rationale for "yet another static site generator" and show with examples how it can be used. The Blog directory contains useful test pages to check processing.

Produced with 'daino' (version Version versionBranch = [0,1,5,3], versionTags = []) from /home/frank/Workspace11/dainoSite/index.md

[4] Initial compilation and linking brings in a large number of packages, e.g. Pandoc, and may take a while; on a typically AMD computer 30..60 Minutes, on a ARM64 (e.g. RaspberryPi4) twice as long for the initial installation.

[5] Producing the pdf files as well with `daino -s` may produce some messages pointing to the latex logs; can usually be ignored.

[6] Currently `settings3.yaml`