

Working with Jekyll and Org-Mode

Most of my notetaking and task management is in org-mode so it makes sense to use that as the basic format of my postings too. This is usually done by publishing a project from org-mode into the location where jekyll keeps its files and let jekyll convert that into something publishable. I'm using a slightly different setup which is more effective for blog posting.

Publishing with org-mode turned out harder than I thought. The publish process is pretty demanding in org-mode and you only end up with raw documents that still need to be processed by jekyll. It occurred to me that <http://github.com> has the ability to render org-mode documents directly and perhaps that library could be used to turn it into a jekyll plugin so I could use the org-mode format directly.

The converter is simple; it uses an org-ruby call to convert the org-file to html and that's it really:

```
module Jekyll
  # Convert org-mode files.
  require 'org-ruby'
  class OrgConverter < Converter
    safe true

    def setup
      # No-op
    end

    def matches(ext)
      ext =~ /org$$/i
    end

    def output_ext(ext)
      ".html"
    end

    def convert(content)
      setup
      Orgmode::Parser.new(content).to_html
    end
  end
end
```

```
end
end
end
```

I can now just write org-mode files with a frontmatter and they'll end up automatically as blog postings. As yml frontmatter needs to come first in the file to make jekyll happy this can't be hidden in an org-mode construct like a comment block or something else that org-mode itself ignores. This makes it harder to use the blog postings for anything else than jekyll because the frontmatter will get in the way; exporting the file to PDF for example. There is obviously room for improvement, but this simple plugin directly gives a workable system.

To have a reference document for writing I created a test org-mode file, with the rendered result here. This file helps to check what org-mode constructs render into something useful and verifying visual layout of them. Not everything worked as I had hoped, but given the amount of complexity that got eliminated I'm quite happy with it.

Issues that I found in the rendering:

- the headers start at level 1 which is probably 1 or 2 levels too high for my purpose; I haven't found a way to correct this yet. I probably should file a feature request for this;
- footnotes do not work, which I would use to keep links nicely at the bottom of an article.
- some rendering is ugly (blockquotes for example), but that's probably not a direct consequence of the org-mode converter
- there are only a couple of org-mode environments supported;
- the use of liquid tags that jekyll uses is somewhat cumbersome.

I was pleasantly surprised by the code highlighting though, which worked out of the box for me.

The next step is finding or making some helper functions in emacs lisp to support working with drafts and publishing.