



Using GitHub Pages Blogging

Sep 6, 2014

After exploring several options for a new programming blog, I settled on GitHub Pages, with Jekyll powering it behind the scenes. Main reasons for this decision were:

- A workflow that's similar to what you would use for your programming projects. Each blog post is treated as a file, which you commit to git and then push.
- Powerful templating and extensibility. There's a lot you can do out of the box, and on top of that you can add lots of new designs, shortcuts and features.
- No database. Pages are static, which makes them very snappy.

That said, the process wasn't that straight-forward for me, despite several blogs and tutorials alluding to how easy it should be (there's probably some Murphy's law about this). The main problem is that there seems to be no comprehensive, end-to-end guide that explains it step by step. GitHub has instructions for setting up GitHub Pages, and at the end it says, "oh by the way, we recommend using Jekyll for writing content here" and links to Jekyll's documentation, which is of course Jekyll-centric. Combining the two together can be an error-prone process. Fortunately, after fiddling with it for a few hours I finally figured it out. I hope this helps someone.

The first step is to setup GitHub Pages. This part is straight-forward: simply create a new repository on github and name it username.github.io, with username being your user name. Then clone it to your computer, add an index.html and you're done with the basic setup. [Github's instructions on this are pretty nice](#).

Once you have your repository cloned to your computer, you need to install Jekyll. Assuming you already have Ruby and Bundler installed, the easiest way to install Jekyll is to create a Gemfile in the root directory and add the github-pages gem to it.

```
source 'https://rubygems.org'
gem 'github-pages'
```

This gem installs Jekyll with some default configuration options used on Github, as well as all of Jekyll's dependencies. Do a bundle install, wait a few minutes and it should be done. If you don't have Ruby or Bundler installed yet, [refer to these more detailed instructions](#).

To make sure Jekyll works, you can go to your Terminal and type:

```
jekyll serve -w
```

And it should start the web server at http://localhost:4000. You will of course need an index.html file (this is a web server, after all), which you can create in your favorite text editor. There's no need to restart Jekyll after creating the file, since the -w parameter tells it to watch for any changes (the feature is called "auto-regeneration").

For me, the troubles started here. The index.html page worked fine (I saw "hello world" when I navigated to http://localhost:4000 on my browser), but it clearly wasn't going to be sufficient for blogging. So I created a new Jekyll project from the root of my repository:

```
jekyll new blog
```

This created [a boilerplate Jekyll directory structure](#) under the /blog folder. Starting the web server from this folder worked fine, but when I committed my changes and pushed them, nothing happened. I then noticed that Github sent

me this email warning:

d idgd

elif ~~if~~ ~~as~~ ~~path~~ ~~and~~ ~~of~~ ~~scre~~ ~~xts~~ ~~siat~~

ble//sth

achy fl


Following the instructions in the link provided – which essentially say “when something goes wrong, run Jekyll locally to troubleshoot problems” – I went to the root folder of the repository and tried to start the Jekyll server, which gave this error:

```
ersoz:egeersoz.github.io egeersoz$ jekyll serve -w
Configuration file: none
      Source: /.../projects/egeersoz.github.io
  Destination: /.../projects/egeersoz.github.io/_site
    Generating...
Build Warning: Layout 'post' requested in blog/_posts/2014-09-06-welcome-to-jekyll.markdown
Build Warning: Layout 'page' requested in blog/about.md does not exist.
Build Warning: Layout 'default' requested in blog/index.html does not exist.
Conversion error: Jekyll::Converters::Scss encountered an error converting 'blog/css/main
jekyll 2.3.0 | Error:  File to import not found or unreadable: base.
Load path: /.../projects/egeersoz.github.io/_sass
```

OK, so what's going on here? The problem seems to be that it could not find some of the CSS partials that main.scss was trying to import, because they were in blog/.sass-cache, which is one folder deep in relation to where the Jekyll service was running. Google'ing this issue didn't result in anything helpful, which usually means (for me at least) that I was doing something wrong.

Anyway, after lots of trial and error, which included playing with the config.yml file and even uninstalling/reinstalling Jekyll and all of its dependencies, I finally said screw it and copied all the files and subfolders inside the blog folder to the root of my repository, and deleted the blog folder. Jekyll stopped complaining, and when I pushed my changes, everything worked fine.

Ruby Zen Garden

Ruby Zen Garden  [egeersoz](#) Tips and tricks for mastering the Ruby programming language.
ege.ersoz@gmail.com