My coding journal About

Moving from Blogger to Jekyll and GitHub Pages

May 11, 2021

Moving from Blogger to Jekyll and GitHub Pages

I wanted something simple that would allow me to use markdown to create posts quickly. I had used Blogger for some of my posts earlier and I thought it would be good to move them to whatever I choose next. After some research I settled on *Jekyll and GitHub Pages*.

Setting up Jekyll

Not much here - installed Jekyll locally so I could easily preview the posts.

```
~$sudo apt-get install ruby-full build-essential zlib1g-dev
~$echo '# Install Ruby Gems to ~/gems' >> ~/.bashrc
~$echo 'export GEM_HOME="$HOME/gems"' >> ~/.bashrc
~$echo 'export PATH="$HOME/gems/bin:$PATH"' >> ~/.bashrc
~$source ~/.bashrc
~$gem install jekyll bundler
```

Creating new site

To create a new Jekyll site

```
~$jekyll new vivekrathod.github.io
~$cd vivekrathod.github.io
~/vivekrathod.github.io$bundle exec jekyll serve
```

This will serve the website at http://localhost:4000

Adding to GitHub

Make sure that the repository on GitHub has Pages enabled - by default if you are creating a per user website then the name of the repository needs to be username.github.io.

```
github.io$git init
github.io$git remote add origin https://github.com/vivekrathod/vivekrathod.githu
github.io$git add .
github.io$git commit -m 'Initial GitHub pages site with Jekyll'
github.io$git push origin master
```

I had to switch to using SSH with GitHub and I had never set it up before

```
~/vivekrathod.github.io$ ssh-keygen -t ed25519 -C "vivekrathod@users.noreply.g
cat ~/.ssh/id_ed25519.pub
```

Add the contents of the public key in GitHub. Then add the key to SSH agent - make sure its running

```
ithub.io$ eval "$(ssh-agent -s)"

ithub.io$ ssh-add ~/.ssh/id_ed25519
ithub.io$ git remote set-url origin git@github.com:vivekrathod/vivekrathod.githu
```

Importing posts from Blogger

First export all posts (Blooger calls it Backup) as XML. Follow official Jekyll documentation at Blogger Import

```
~/vivekrathod.github.io$ gem install jekyll-import
```

```
$ ruby -r rubygems -e  'require "jekyll-import"; JekyllImport::Importers::Blog
```

Using custom domain name with GitHub pages

I wanted to use my domain name vivekrathod.com for this blog. I followed the documentation on GitHub Pages Under `Settings` and then `Pages` I put `vivekrathod.com` as custom domain. Once I did that GitHub Pages does an automatic check if the DNS records for this domain are properly set. If not, you are told what to do. I had to add a `A` record in my DNS server to point to GitHub Pages IPs

```
~/vivekrathod.github.io$ dig vivekrathod.cOM +noall +answer
vivekrathod.cOM.        0       IN      A       185.199.109.153
vivekrathod.cOM.        0       IN      A       185.199.110.153
vivekrathod.cOM.        0       IN      A       185.199.108.153
vivekrathod.cOM.        0       IN      A       185.199.111.153
```

As recommended, I also added `CNAME` record pointing `www.vivekrathod.com` to `vivekrathod.github.io.` Once you do this, give it some time and re-check using the same page or use the `dig` command to see if the domain points to correct IPs. GitHub Pages also initiates generation of SSL/TLS certificate for enabling HTTPS. I chose to enforce HTTPS.

Enabling comments on posts

This one was interesting - Jekyll's default theme *minima* apparently supports *disqus* for commenting. I wanted something less invasive so after some research I chose *utteranc.es* Getting it to work turned out to be a bit of a challange for me. The default *minima* theme is a gem based theme and it has it's own *_includes* and *_layouts* directories so I was not sure where I would put the JavaScript for enabling comments using *utteranc.es*. I could override the *default* or the *post* layout (*default.html* or *post.html* from *_layouts*) by copying from the *minima* directory

```
~/vivekrathod.github.io$ bundle info minima
  * minima (2.5.1)
        Summary: A beautiful, minimal theme for Jekyll.
        Homepage: https://github.com/jekyll/minima
        Path: /home/vivekrathod/gems/gems/minima-2.5.1
~/gems/gems/minima-2.5.1$ ls
LICENSE.txt  README.md  _includes  _layouts  _sass  assets
~/gems/gems/minima-2.5.1/_layouts$ ls
default.html  home.html  page.html  post.html
```

But then I would loose the ability to get updated layouts and/or includes when I update *minima*. Luckily, I noticed that *minima* supports *disqus* comments. I found this in the *post.html*

```
  {%- if site.disqus.shortname -%}
    {%- include disqus_comments.html -%}
  {%- endif -%}
```

So I added a fake *disqus shortname* in the *_config.yml*

```
#specifying a fake disqus shortname here and overriding _includes/disqus_comme
disqus:
    shortname: utteranc.es
```

And this way I could load the comments for all posts and still be able to update *minima*

Creating new posts

I found a Rakefile from the Jekyll Bootstrap project that simplifies creating a new posts. I removed some things in it I dont need, and it allows me to simply do this `rake post title="A Title" [date="2012-02-09"] [tags=[tag1,tag2]] [category="category"]`

My coding journal

My coding journal
vivek.rathod@gmail.com

vivekrathod
vrthd

Intended to be a place where I can capture coding and programming knowledge. It is mostly focused on .NET and C# with some work related to databases and security as well.