

Feb 08 2023

Add a post!

It's pretty easy. And cheaper than shelter!

0. Prerequisites

You'll need git on the command line, and a Java 17+ VM installed:

```
$ java -version
java -version
openjdk version "17.0.2" 2022-01-18
OpenJDK Runtime Environment (build 17.0.2+8-86)
OpenJDK 64-Bit Server VM (build 17.0.2+8-86, mixed mode, sharing)
```

(Let me know if a tutorial on how to install/select JVMs would help!)

1. Clone the **d24n-unstatic** repo

```
$ git clone git@github.com:d24n/d24n-unstatic.git
git clone git@github.com:d24n/d24n-unstatic.git
Cloning into 'd24n-unstatic'...
Enter passphrase for key '/home/swaldman/.ssh/id_ed25519':
remote: Enumerating objects: 1394, done.
remote: Counting objects: 100% (1394/1394), done.
remote: Compressing objects: 100% (413/413), done.
remote: Total 1394 (delta 472), reused 1309 (delta 388), pack-reused 0
Receiving objects: 100% (1394/1394), 4.87 MiB | 29.33 MiB/s, done.
Resolving deltas: 100% (472/472), done.
```

2. Create your "untemplate"

The blog's source — html, markdown, scala, static resources — all live in the `d24n` directory of your new repository.

```
d24n
+ - src/          (Scala/Java source code)
+ - static/       (static resources of the website)
+ - untemplate/   (html and markdown templates that define the generated con
```

Blog entries live in untemplate, under `untemplate/org/d24n/site/mainblog`.

- Anything (recursively) under this directory that looks like `entry-whatever.md.untemplate` will be considered a template for a markdown blog entry.
- Replace the `.md.untemplate` suffix with `.html.untemplate` for an HTML template, etc.
- The `whatever` part in the middle is pretty arbitrary, though I've mostly used post titles.
- As a matter of pure convenience, I've organized the `mainblog` directory by year, so this post (yours too!) should go in `year2023`.

Note: This directory hierarchy is going to map to a JVM package hierarchy, so I could not have named this directory simply `2023`, because that would not be a valid *identifier*.

Use your favorite editor to create a file in `untemplate/org/d24n/site/mainblog/year2023`.

From the repository root (`d24n-unstatic`) directory...

```
$ emacs d24n/untemplate/org/d24n/site/mainblog/year2023/entry-hi-from-steve.m
```

You'll want the file to begin with some annoying boilerplate. I've made it available to copy and paste [on the site itself](#). Or, here it is!

```
> val UntemplateAttributes = immutable.Map[String,Any] (
>   "Title"    -> "Activities in 2018",
>   "Author"   -> "Chris Peel",
>   "PubDate"  -> "2019-04-17",
> )
>

given PageBase = PageBase.fromPage(input.renderLocation)
```

```
(input : D24nSite.MainBlog.EntryInput)[]~()>      ### modify Title/Author/Pubdate above, add markdown or html below
```

Now, just do as the **delimiter comment** instructs. Modify Title/Author/Pubdate above, add markdown (since this is a **.md.untemplate** file) below!

```
> val UntemplateAttributes = immutable.Map[String,Any] (  
>   "Title"    -> "Hi from Steve!",  
>   "Author"   -> "Steve Waldman",  
>   "PubDate"  -> "2023-02-08",  
> )  
  
given PageBase = PageBase.fromPage(input.renderLocation)  
  
(input : D24nSite.MainBlog.EntryInput)[]~()>      ### modify Title/Author/Pubdate above, add markdown or html below  
  
Hi! My name is Steve, and I'm, well, _disagreeable_. It's lovely to meet you in any case, although  
if you've any sense that's not reciprocal, because:  
  
* I am stinky.  
* My head is either a porcupine or a bowling ball, depending on the day.  
* I write gratuitous lists.  
  
## The Early Years  
  
I was disagreeable then too!
```

I'll slip this in as the post **below** this one, so you can see how it renders!

Note: Don't write your post title above your text, just include it in **UntemplateAttributes**. The blog will render its posts' titles.

3. Generate and browse the static site

Just go to the root of the repository (i.e. the **d24n-unstatic** directory), and type:

```
$ ./d24n-site-gen
```

You should see a report showing what pages got generated and what static directories got copied.

You should have a file-system browsable website under the **docs/** directory of the repository now. Point your browser at **docs/index.html**.

If you are on a Mac, you can just type

```
$ open docs/index.html
```

Note: If you accidentally just type `./d24n-site`, the site will be generated, but into the more conventional `public` directory rather than `docs`. We are generating (for now) into `docs` because our temporary home GitHub Pages serves from that directory. You can just `rm -rf public`, or use type `./veryclean.sh` from the root of your repository directory if this happens.

4. Tweak!

You can, of course, look at your generated post, edit your untemplate file, regenerate again with `./d24n-site-gen`, reload, rinse, repeat.

But this gets a smidgen cumbersome.

If you use the command

```
./d24n-site-devcycle
```

You'll start up a server on port 8999 of your local machine.

Because (temporarily), this site lives on *GitHub Pages* under the path `d24n_unstatic`, you'll need to point your browser to `http://localhost:8999/d24n-unstatic/` to see the served site.

Once the local development site is running, you can browse to your permalink, edit your untemplate, hit save, then refresh your page after 5-10 seconds to admire and critique your update.

5. Publish

Once you have your post looking lovely, hit `<ctrl-c>` to kill the development server, then

```
$ ./d24n-site-gen
```

to regenerate the static site. Then

```
$ git add .  
$ git commit -m "Added an effing brilliant new post!"  
$ git push
```

and within a few minutes you should have it live at <https://d24n.github.io/d24n-unstatic/>. Hooray!

Give it a try!

Published by Steve Waldman on 2023-02-08

Decentralization Foundation

4701 Patrick Henry Dr

Building 25

Santa Clara CA, 95054

decentralizationfoundation@gmail.com

Connect

Join our [email list](#)

Read our [RSS Feed](#)

Connect on [meetup.com](#)

Follow us on Twitter: [@d24nOrg](#)

Powered by [unstatic](#), [untemplates](#), [tapir](#), [zio](#), [flexmark-java](#), [jsoup](#), [mill](#), and [scala 3](#) *inter alia*.