



# Safe P2P Browsing Web With Beaker and Dat

POST

B!

LINE

 Beaker Browser

Recently I have been searching for a way to host websites in peer-to-peer manner. I had a quick look into Tor/Onion and I2P. They don't instill lots of inspiration for non-tech users so I kept looking. I came across [Beaker browser](#) and [Dat protocol](#) that make browsing and hosting P2P websites simple and easy. This post will give you brief intro into Beaker and Dat so you can quickly embark and launch your own Dat website.

## Motivation

---

I read Beaker browser website and found motivation behind their product:

1. Avoid vendor lock-in and dependency on centralised infrastructure providers (Amazon AWS, Microsoft Azure, Google App Engine)
2. Have peer-to-peer network where users own their data and can run independently
3. Make site hosting seamless and built-in into your browser
4. Provide integrity and versioning to websites
5. Allow anyone to clone and keep history of websites
6. Make sharing easy and still keep URI semantics of open web

I will dig deep into the Dat protocol and write about it in another post.

## Guarantees by Beaker Browser & Dat protocol

---

### 1. Browsing privacy: No

- Anyone having knowledge of a `dat://[0-9a-f]{64}` URL can see which devices also accessing the Dat URL

### 2. Data privacy: Yes

- Dat websites use sandboxes and can't export user data
- Use thick application model (store locally using `localStorage`, `indexedDB` & Dat APIs). Read more at <https://beakerbrowser.com/docs/inside-beaker/thick-applications.html>

### 3. Secret sharing: Yes

- Requested URLs are hashed via HMAC and peers who don't know the URL can't make sense of it
- Dat website URL is a 64 hex character string, equivalent to 32 byte key/256 bit key. It's used as symmetric encryption key for that URL

### 4. Site integrity: Strict Content Security Policy (CSP)

- Only assets from the sandbox are executed, there's no hot-linking or loading 3rd party scripts & images. This prevents ad-trackers

Read details at <https://beakerbrowser.com/docs/inside-beaker/privacy-and-security.html> .

## Hosting on your own server

---

By default you can rely on Beaker browser to host Dat websites for you. However, once you put your PC in sleep mode or turn it off, the site will be gone if it's not cached by any peer. Fortunately, you can host Dat websites outside Beaker browser using `dathttpd`. For details, read the official guide:

- <https://beakerbrowser.com/docs/tutorials/host-outside-of-beaker.html>
- `dat://beakerbrowser.com/docs/tutorials/host-outside-of-beaker.html`
- <https://github.com/beakerbrowser/dathttpd>

Feel free to host Dat websites on a Linode box, or run your own hardware server.

## Enable domain for Dat websites

---

Now you have seen the magic of Dat protocol and simplicity of Beaker browser. There's still a pain point: No one can remember the actual link because no genius can memorize 64 hex characters. This is where engineering minds shine.

- In order to use `my-domain.tld` in Beaker browser, you must be able to plant the file `https://my-domain.tld/.well-known/dat`
- The content of the file must contain Dat URL and TTL (time to live in seconds)
- Name resolution is handled transparently

Here's the actual example file provided by makers of Beaker browser:

<https://beakerbrowser.com/.well-known/dat>

```
dat://87ed2e3b160f261a032af03921a3bd09227d0a4cde73466c17114816cae43336/  
ttl=3600
```

This precious file allows you to access the public key above as `dat://beakerbrowser.com` painlessly. For more

information, read:

- <https://github.com/beakerbrowser/beaker/wiki/Authenticated-Dat-URLs-and-HTTPS-to-Dat-Discovery>

**There's a gotcha:** Your Dat website doesn't need to be in sync with your HTTPS version. There's no way to enforce it at the moment.

**NOTE:** Make sure the folder of your Dat website always has its meta-data file `dat.json` :

```
{
  "url":
  "dat://3fa25efcd82c180934b69503b718f56856385026279cdec5120a53c4eb48df6b/",
  "title": "visc.network",
  "description": "Vietnam's Initiative for Smart Contracts"
}
```

**NOTE:** If you use Hugo to generate web pages to be hosted with HTTP server and Dat, make sure to:

- Set `canonifyURLs` to `false` in `config.toml`
- Run command `hugo -b '/' -d _dsite` to set base URL to `/`
- Check <https://github.com/vietlq/vietlq.github.io/tree/source> for inspiration



## References

---

I hope this was useful and you can try hosting your own Dat websites. Get started with these links:

- <https://beakerbrowser.com/docs/tutorials/create-a-blog.html>
- <https://beakerbrowser.com/docs/apis/>
- `dat://taravancil.com/`
- `dat://taravancil.com/blog/how-i-publish-taravancil-com/`
- <https://github.com/taravancil/taravancil.com>

## Happy hacking

---

If you follow the above links carefully, you will be able to host your own Dat site with nice domain name like I did:

`dat://simpleit.us/`