

Un navigateur pour diffuser votre site web en pair à pair

Les technologies qui permettent la décentralisation du Web suscitent beaucoup d'intérêt et c'est tant mieux. Elles nous permettent d'échapper aux silos propriétaires qui collectent et monétisent les données que nous y laissons.

Vous connaissez probablement Mastodon, peerTube, Pleroma et autres ressources qui reposent sur le protocole activityPub. Mais connaissez-vous les projets [Aragon](#), [IPFS](#), ou [ScuttleButt](#) ?

Aujourd'hui nous vous proposons la traduction d'un bref article introducteur à une technologie qui permet de produire et héberger son site web sur son ordinateur et de le diffuser sans le moindre serveur depuis un navigateur.

[L'article original](#) est issu de [la série Dweb](#) (Decentralized Web) publiée sur [Mozilla Hacks](#), dans laquelle [Dietrich Ayala](#) met le projecteur sur toutes les initiatives récentes autour du Web décentralisé ou distribué.

Traduction Framalang : bengo35, goofy

Blue Link Labs et Beaker

par Tara Vancil



Nous sommes [Blue Link Labs](#), une équipe de trois personnes qui travaillent à améliorer le Web avec [le protocole Dat](#) et un navigateur expérimental pair à pair qui s'appelle [Beaker](#).



L'équipe Blue Link Labs

Nous travaillons sur Beaker car publier et partager est l'essence même du Web. Cependant pour publier votre propre site web ou seulement diffuser un document, vous avez besoin de savoir faire tourner un serveur ou de pouvoir payer quelqu'un pour le faire à votre place.

Nous nous sommes donc demandé « Pourquoi ne pas partager un site Internet directement depuis votre navigateur ? »

Un protocole pair-à-pair comme `dat://` permet aux appareils des utilisateurs ordinaires d'héberger du contenu, donc nous utilisons `dat://` dans Beaker pour pouvoir publier depuis le navigateur et donc au lieu d'utiliser un serveur, le site web d'un auteur et ses visiteurs l'aident à héberger ses fichiers. C'est un peu comme BitTorrent, mais pour les sites web !

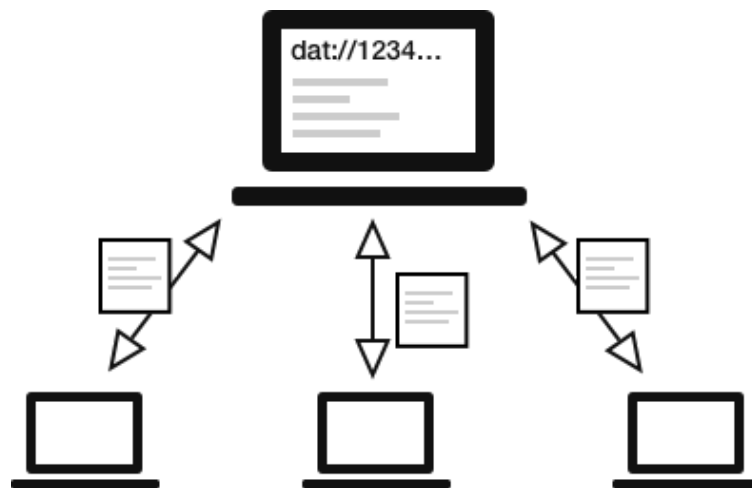
Architecture

Beaker utilise un réseau pair-à-pair distribué pour publier

des sites web et des jeux de données (parfois nous appelons ça des «dats»).

Les sites web `dat://` sont joignables avec une clé publique faisant office d'URL, et chaque donnée ajoutée à un site web `dat://` est attachée à un log signé.

Les visiteurs d'un site web `dat://` peuvent se retrouver grâce à une *table de hachage distribuée*¹, puis ils synchronisent les données entre eux, agissant à la fois comme *téléchargeurs* et *téléverseurs*, et vérifiant que les données n'ont pas été altérées pendant le transit.



Une illustration basique du réseau `dat://`

Techniquement, un site Web `dat://` n'est pas tellement différent d'un site web `https://`. C'est une collection de fichiers et de dossiers qu'un navigateur Internet va interpréter suivant les standards du Web. Mais les sites web `dat://` sont spéciaux avec Beaker parce que nous avons ajouté une API (interface de programmation) qui permet aux développeurs de faire des choses comme lire, écrire, regarder des fichiers `dat://` et construire des applications web pair-à-pair.

Créer un site Web pair-à-pair

Beaker rend facile pour quiconque de créer un nouveau site web dat:// en un clic ([faire le tour des fonctionnalités](#)). Si vous êtes familier avec le HTML, les CSS ou le JavaScript (même juste un peu !) alors vous êtes prêt·e à publier votre premier site Web dat://.

Les développeurs peuvent commencer par regarder la documentation de notre [interface de programmation](#) ou [parcourir nos guides](#).

L'exemple ci-dessous montre comment fabriquer le site Web lui-même via la création et la sauvegarde d'un fichier JSON. Cet exemple est fictif mais fournit un modèle commun pour stocker des données, des profils utilisateurs, etc. pour un site Web dat:// : au lieu d'envoyer les données de l'application sur un serveur, elles peuvent être stockées sur le site web lui-même !

```
// index.html
Submit message
<script src="index.js"></script>

// index.js
// first get an instance of the website's files
var files = new DatArchive(window.location)
document.getElementById('create-json-button').addEventListener('click', saveMessage)
async function saveMessage () {
var timestamp = Date.now()
var filename = timestamp + '.json'
var content = {
timestamp,
message: document.getElementById('message').value
}

// write the message to a JSON file
```

```
// this file can be read later using the DatArchive.readFile
API
await files.writeFile(filename, JSON.stringify(content))
}
```

Pour aller plus loin

Nous avons hâte de voir ce que les gens peuvent faire de dat:// et de Beaker. Nous apprécions tout spécialement quand quelqu'un crée un site web personnel ou un blog, ou encore quand on expérimente l'interface de programmation pour créer une application.

Beaucoup de choses sont à explorer avec le Web [pair-à-pair](#) !

- [Pour Beaker, suivez le guide !](#)
- Tara Vancil en conférence au JSConf EU 2018 : (YouTube) [A Web Without Servers](#)
- [Documentation de Beaker](#)
- [Documentation de DAT](#)
- [p2pforever.org](#) – un modeste carrefour de ressources pair-à-pair sur le Web
- [Beaker sur GitHub](#)
- [Installer Beaker Browser](#)
 - et parcourir l'article ci-dessus en DAT :
dat://12420d10773ccc076898b7d782263cde666112b74dff6d4e1eabf2e4bfcdb672/

Documentation plus technique

- [How Dat works](#), un guide en anglais qui expose tous les détails sur le stockage des fichiers avec Dat
- [The Dat Protocol Book](#), également en anglais, plus complet encore.

À propos de [Tara Vancil](#)

Tara est la co-créatrice du navigateur Beaker. Elle a

travaillé précédemment chez Cloudflare et participé au [Recurse Center](#).