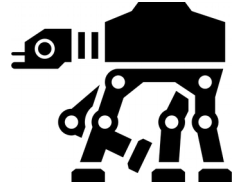


Publier automatiquement sur les réseaux sociaux pour ne plus perdre de temps... : voici mes astuces !



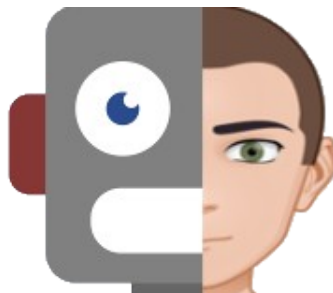
V2.5.3 mise à jour le 27/07/2018
(v1.0 publiée le 31/07/2017)

Table des matières

| | |
|---|----|
| Introduction..... | 2 |
| Principe général..... | 2 |
| Avertissements..... | 3 |
| Notes de versions (majeures uniquement)..... | 3 |
| Schéma d'ensemble du projet..... | 4 |
| Matériel utilisé..... | 4 |
| Sources m'ayant aidé à la configuration..... | 4 |
| Étape n°1 : générer un fil RSS de son travail de veille..... | 5 |
| Solution « je suis autonome » avec mon TTRSS..... | 5 |
| Solution simple quand on n'a pas de serveur « perso »..... | 5 |
| Étape n°2 : installer les outils de publication depuis ce fil RSS vers Mastodon..... | 6 |
| Le système d'exploitation et les logiciels essentiels..... | 6 |
| L'installation et la configuration de feed2toot..... | 6 |
| Installer le « boost » automatique en fonction de hashtags spécifiques..... | 7 |
| Configurer des toots automatiques !..... | 9 |
| Étape n°3 : installer les outils de publication depuis ce fil RSS vers Diaspora*..... | 10 |
| Étape n°4 : installer les outils de publication depuis ce fil RSS vers Twitter..... | 11 |
| Étape n°5 : enjoy !..... | 12 |
| Quels gains au final ?..... | 12 |
| Quels points de vigilance ?..... | 12 |

Auteurs de ce documents :

- Frédéric Véron (Alias Sangokuss ou SVTux), initiateur du document,
- Samuel Vermeulen (SVNET Libre), contributeur.



« Voici mon #bot ! Et le vôtre, il ressemble à quoi ? »

Introduction

Si comme moi vous êtes présent sur différents réseaux sociaux, vous savez qu'il s'agit d'une activité intéressante mais chronophage. C'est dans une situation comme celle-ci que le monde des robots peut commencer à nous aider ! Mais pour automatiser votre veille et votre communication sur ces derniers, plusieurs questions se posent :

- Quels outils (bots) existent ? Lesquels utiliser ? Pour quels réseaux sociaux ?
- Ces outils sont-ils respectueux de mes données ? Qui aura accès à ma veille ? A mes publications ?
- Comment différencier ses publications automatiques (bots), ponctuelles et personnelles en fonction des réseaux sociaux ?
- Comment répondre à ces questions si vous utilisez des réseaux sociaux libres et décentralisés, le plus souvent ignorés des outils propriétaires de publication ?

Voici les solutions libres que j'utilise. Mon aide-mémoire technique en quelque sorte. L'objectif est ici, du moins je l'espère, de vous permettre de découvrir, tester et mieux comprendre comment j'ai répondu à ces questions en fonction de mes besoins.

Bonne lecture !

Principe général

L'essentiel de mon travail de veille passant par mon agrégateur de flux RSS (et oui, on ne fait rien de mieux pour qui recherche une information brute, non triée par un robot et... surtout qui respecte ma vie privée), il s'agissait dans mon cas personnel de répondre au cahier des charges suivant :

- **en un clic** unique, de rediffuser / publier les informations lues sur mon agrégateur RSS sur l'ensemble des réseaux sociaux publics auxquels je suis inscrit, à savoir : Mastodon, Diaspora* et Twitter. Cette contrainte est « forte et difficile à mettre en oeuvre » dans le sens où aucun outil propriétaire ne le permet.
- **Automatiser (bot ou robot) la publication de certains messages** (par exemple certaines informations institutionnelles telles que les lettres de rentrées du ministère...), soit sur la base du « retweet » ou soit sur l'indexation d'une liste de flux RSS.
- De pouvoir dans certaines situations **publier des réponses automatiques**.
- Conserver la **possibilité de publier manuellement en cas de besoin** (par exemple pour échanger, répondre à des questions, interpellé... « à l'ancienne »).

En bref, il s'agissait pour moi de limiter au maximum ma perte de temps dans la publication afin de pouvoir, si nécessaire, être plus réactif et efficace dans les échanges (ou d'autres tâches du quotidien ! CQFD), tout en garantissant l'intégrité et la confidentialité de mes données. En effet, il m'est impensable d'accepter qu'un outil propriétaire ait accès aux données de mon travail de veille sur l'ensemble des réseaux sociaux. Tout en permettant de séparer et sécuriser les données de ma veille personnelle de celle de ma veille publique.

Avertissements

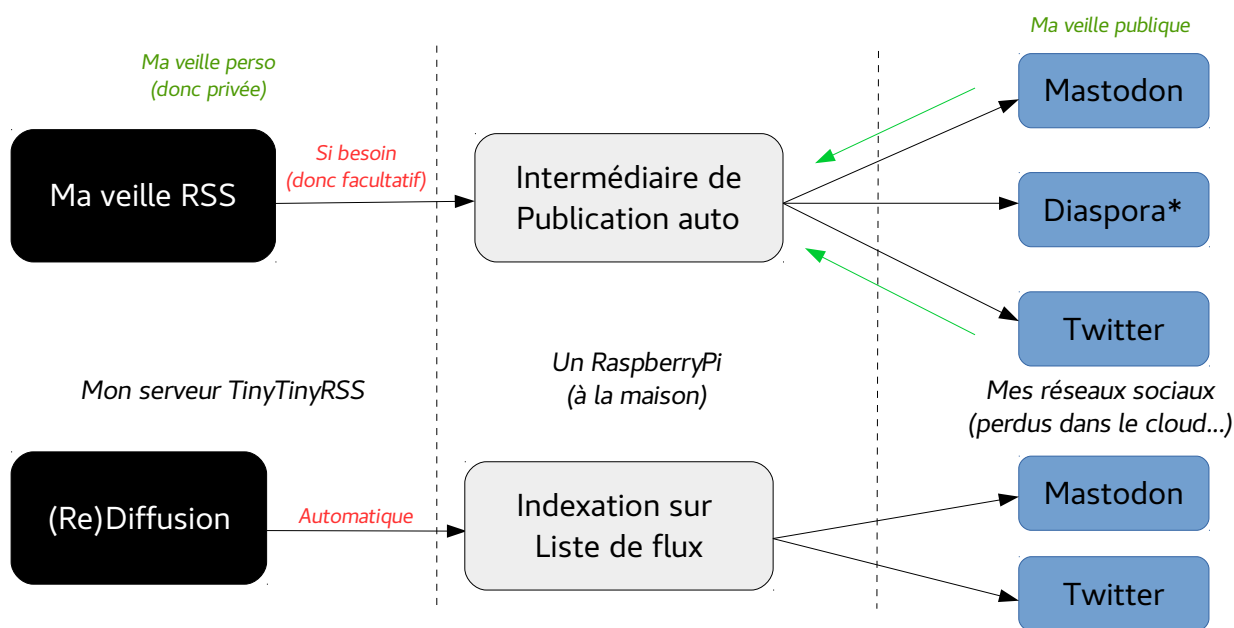
Alors, oui, c'est technique. Mais si j'y suis arrivé plutôt simplement et rapidement, et pour celui qui possède un minimum de compétence numérique, je ne doute pas que mon aide mémoire suffira amplement pour y parvenir. Et puis, nul doute qu'un outil « clé en main » (pourquoi pas avec une interface graphique...) finira bien par voir le jour !

Ce document a donc été pour moi un investissement sur le long terme afin de me libérer du temps de cerveau disponible pour des choses bien plus passionnantes que de passer ma vie sur les réseaux sociaux... !

Notes de versions (majeures uniquement)

- V2.5.3 (juillet 2018) :
 - Modification du nombre de caractères utilisés pour feed2tweet (remplacement des 140 caractères par 250).
 - Contribution de Samuel Vermeulen au document (feed2toot et feed2tweet en particulier).
- V2.4 (juin2018) :
 - Indexation des publications automatiques sur la base d'une listes de flux de confiance.
 - Modifications mineures de la rédaction.
- V2.0 (Octobre 2017) :
 - Mastodon : ajout de rediffusions automatiques en fonction d'éléments prédéterminés.
 - Mastodon : ajout de réponses interactives automatiques du bot (via déclencheur).
 - Twitter : ajout du principe des sites TweetDeck et IFTTT qu'il est possible de d'utiliser/détourner.
- V1.0 (juillet 2017) : version initiale n'utilisant que la rediffusion de RSS marqués vers
 - Twitter, Mastodon et Diaspora*.

Schéma d'ensemble du projet



Matériel utilisé

- Un Rpi B (*machine sur laquelle les automatisations seront configurées*)
 - Validation de l'ensemble du tuto sur Raspbian Jessie (le présent document devra donc faire l'objet d'une mise à jour pour Stretch).
- Un serveur TinyTinyRSS (*soit le vôtre si vous en avez un, soit un serveur public tel « Framanews »*).
- Mon cerveau ! (*au début, et celui du bot... à la fin!*)

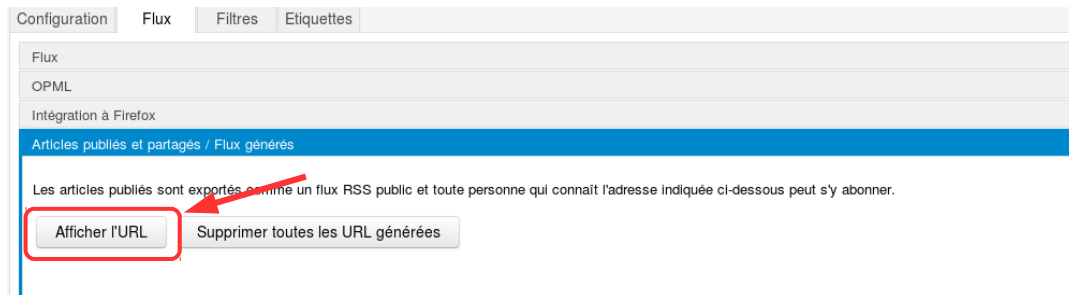
Sources m'ayant aidé à la configuration

- Le site <https://www.raspberrypi.org/>
- Le site <https://tt-rss.org/>
- Le blog de Carl Chenet <https://carlchenet.com/vos-flux-rss-sur-le-reseau-mastodon-avec-le-bot-feed2toot/>
- Le site de FeedDiasp <https://debakel.github.io/feedDiasp/> et <https://github.com/celisoft/feedDiasp>
- Feed2Tweet <https://feed2tweet.readthedocs.io/en/latest/index.html>
- Poster des toots automatiquement : <https://carlchenet.com/automatically-send-toots-to-the-mastodon-social-network/>
- Site de TweetDeck <https://tweetdeck.twitter.com/> (à utiliser avec modération si la vie privée est essentielle pour vous)
- Le site IFTTT <https://ifttt.com> (à utiliser avec modération si la vie privée est essentielle pour vous)

Étape n°1 : générer un fil RSS de son travail de veille.

Solution « je suis autonome » avec mon TTRSS

Pour générer le flux RSS de mon travail de veille, j'utilise simplement l'application TinyTinyRSS (agrégateur de flux RSS) installée sur mon serveur Yunohost (mon auto-hébergement). Rien de plus simple en allant dans l'espace de configuration :



Il me suffit ensuite, quand je fais mon travail de veille RSS de cliquer sur « publier » pour que le flux se retrouve sur page ! A la fin de ce tuto, ce simple clic suffira pour publier sur mes réseaux sociaux.



Solution simple quand on n'a pas de serveur « perso »

Comme j'imagine que certains d'entre vous n'ont pas nécessairement envie (ou les compétence pour) de mettre en place un serveur TinyTinyRSS, le service [Framanews](#) de Framasoft, basé sur TinyTinyRSS fera la même chose.



Étape n°2 : installer les outils de publication depuis ce fil RSS vers Mastodon.

[Changement v2.x] Depuis le 21/10, j'ai ajouté au Raspberry Pi B quelques fonctionnalités de publications automatiques. L'idée générale étant d'avoir une forme de publication automatique sur des sujets (ou # spécifiques) de manière différenciée selon les réseaux.

Notez dès à présent que l'outil que nous allons installer (« feed2toot », pour ne pas le citer) est hautement personnalisable et offre donc des possibilités vraiment intéressantes que je vous laisse explorer par vous-même.

Le système d'exploitation et les logiciels essentiels

1. Installer Raspbian sur une carte SD / micro-SD (*commande à adapter à votre système Gnu/Linux*).
 - ~\$ su -c 'dd bs=1M if=xxxx of=xxxx'
2. Une fois l'installation terminée, brancher la carte sd sur votre Rpi et démarrer ce dernier.
3. N'oubliez pas de créer un dossier ssh dans le /boot, sinon vous ne pourrez y accéder via ssh...
4. Faire les mises à jour.
 - ~\$ sudo apt-get update
 - ~\$ sudo apt-get upgrade
 - ~\$ sudo apt-get dist-upgrade
 - ~\$ sudo reboot
5. Installer Python :
 - ~\$ sudo apt-get install python3-pip

L'installation et la configuration de feed2toot

- -\$ sudo pip3 install --upgrade setuptools
- -\$ sudo pip3 install cffi
- ~\$ sudo pip3 install feed2toot
- ~\$ sudo adduser --home /var/lib/feed2toot --gecos "" feed2toot
- ~\$ sudo mkdir -p /etc/feed2toot/credentials /var/lib/feed2toot/
- ~\$ sudo chown -R feed2toot:root /etc/feed2toot /var/lib/feed2toot
- ~\$ register_feed2toot_app (*Attention, la double authentification n'est pas prise en charge !*)
- Ce qui aura pour effet d'ajouter l'autorisation sur Mastodon :

| | | | |
|-----------|--|------------------------|--------------|
| feed2toot | lire les données de votre compte poster en tant que vous s'abonner, se désabonner, bloquer et débloquer des comptes | 21 oct. 2017, 10:21 | ✕ Annuler |
|-----------|--|------------------------|--------------|

- ~\$ sudo cp feed2toot_usercred.txt feed2toot_clientcred.txt /etc/feed2toot/credentials/
- ~\$ sudo nano /etc/feed2toot/feed2toot.ini (*et y coller et adapter le contenu ci-dessous*)

```
[mastodon]
instance_url=https://mastodon.social
user_credentials=/etc/feed2toot/credentials/feed2toot_usercred.txt
client_credentials=/etc/feed2toot/credentials/feed2toot_clientcred.txt
```

```
[cache]
cachefile=/var/lib/feed2toot/feed2toot.db
```

```
[rss]
url_list=/etc/feed2toot/rsslist.txt
tweet= [#Actu] « {title} » {link}
```

```
[hashtaglist]
several_words_hashtags_list=/etc/feed2toot/hashtags.txt
```

- Ensuite, si vous devez suivre et publier à partir de plusieurs vous pouvez créer un fichier rsslist.txt qui contiendra la liste des RSS à suivre et à publier automatiquement et y copier-coller vos RSS :
 - ✓ ~\$ mkdir /home/pi/feed2toot
 - ✓ ~\$ nano /home/pi/feed2toot/rsslist.txt
- Il faudra alors remplacer la ligne `url=` de votre fichier de configuration par la ligne suivante :
 - ✓ `uri_list=/home.pi/feed2toot/rsslist.txt`

Remarque : si lorsque vous construisez votre fichier liste de RSS, que vous testez la commande `~$ sudo feed2toot --populate-cache -c /etc/feed2toot/feed2toot.ini` et que des erreurs de « tag » sont renvoyées, c'est que le (ou les) flux choisis sont inadaptés.

Enfin :

- ~\$ chmod 640 /etc/feed2toot/feed2toot.ini

A présent feed2toot est en place, mais il faut automatiser la tâche via cron :

- ~\$ crontab -e (personnellement, par habitude, je choisis l'éditeur nano)
- Et ajouter une ligne telle que :


```
*/15 * * * * feed2toot -c /etc/feed2toot/feed2toot.ini
```

Remarque : ici, le 15 indique que la tâche sera effectuée toutes les 15min ! (perso, je préfère 5 ou 10 min car cela évite de poster trop de messages d'un coup !) **et limiter la réception à 1 item par flux RSS avec l'option -l égale 1.**

Concrètement, lorsque désormais je lis un flux RSS, dans TinyTinyRSS (depuis l'interface web ou l'application dédié sur mon Android), et que je marque un article comme « publié », il sera publié automatiquement sur Mastodon quelques minutes plus tard (bonus : cela permet de pouvoir corriger si erreur).

Très important : afin d'éviter de polluer votre fil par un tsunami de toots, n'oubliez pas de faire la commande suivante :

```
~$ sudo feed2toot --populate-cache -c /etc/feed2toot/feed2toot.ini
```

Cela permet d'intégrer à la base de données les articles déjà présents dans les flux ! (cette fonctionnalité est à utiliser avec prudence, car j'ai déjà rencontré un bug au niveau de la BDD avec feed2toot engendrant une publication incontrôlée...).

Installer le « boost » automatique en fonction de hashtags spécifiques

Principe : nous allons mettre en place le bot Boost afin de booster automatiquement les toots d'un compte spécifique lorsque ce dernier utilise un hashtag spécifique.

- ✓ ~\$ sudo pip3 install boost

Nous allons créer un utilisateur dédié sur le système avec les répertoires `/etc/boost` pour les éléments de configuration et `/var/lib/boost` pour les données :

- ✓ ~\$ sudo adduser --home /var/lib/boost --gecos "" boost
- ✓ ~\$ sudo mkdir -p /etc/boost/credentials /var/lib/boost
- ✓ ~\$ sudo chown -R boost:root /etc/boost /var/lib/boost

Générons maintenant l'autorisation de notre application afin qu'elle puisse utiliser notre compte Mastodon :

- ✓ ~\$ sudo register_boost_app
- ✓ ~\$ sudo cp boost_usercred.txt boost_clientcred.txt /etc/boost/credentials

Il suffit de répondre aux questions qui sont posées par l'application. Si les éléments de configuration sont corrects, deux fichiers sont générés qui contiennent nos autorisations, boost_usercred.txt et boost_clientcred.txt dans votre répertoire courant.

Nous écrivons maintenant la configuration de Boost :

```
[mastodon]
users_to_boost=carlchenet@mastodon.social
instance_url=https://mastodon.social
user_credentials=/etc/boost/credentials/boost_usercred.txt
client_credentials=/etc/boost/credentials/boost_clientcred.txt

[boost]
only_if_hashtags=Mastodon

[sqlite]
sqlitepath=/var/lib/boost/boost.db
```

Pour information :

- Dans la section **[mastodon]** :
 - **users_to_boost** est une liste des comptes Mastodon des utilisateurs intéressants, séparés par une virgule
 - **instance_url** est l'URL vers votre instance Mastodon
 - **user_credentials** est le chemin vers le fichier boost_usercred.txt, généré par la commande register_boost_app
 - **client_credentials** est le chemin vers le fichier boost_clientcred.txt, généré par la commande register_boost_app
- Dans la section **[boost]** :
 - **only_if_hashtags** définit une liste de mot-clé (sans le # devant) séparés par une virgule qui identifieront les toots à booster
- Dans la section **[sqlite]** :
 - **sqlitepath** est un chemin vers le fichier de base de données qui contient l'identifiant des toots déjà boostés.

Notre configuration de Boost est maintenant prête. Si nous voulons éviter d'automatiquement booster les derniers 20 toots correspondant aux critères définis, nous allons lancer pour la première fois Boost avec l'option de la ligne de commande --populate-database :

- ✓ ~\$ boost --populate-database --config /etc/boost/boost.ini

La prochaine exécution de Boost avec la commande suivante boostera automatiquement tous les toots correspondant à nos critères :

- ✓ ~\$ boost --config /etc/boost/boost.ini

Pour nous éviter de lancer cette commande à la main, nous ajoutons une ligne dans notre crontab pour exécuter le Boost à intervalle régulier, comme pour feed2toot ci-dessus.

Configurer des toots automatiques !

- ✓ ~\$ sudo pip3 install toot
- ✓ ~\$ toot login
- ✓ ~\$ nano /home/pi/mon-message.sh
- ✓ Et y insérer :

```
#!/bin/bash  
toot post -m /home/pi/mon-image.png $'le texte de mon message...'  
exit 0
```

- ✓ ~\$ sudo chmod u+x /home/pi/mon-message.sh
- ✓ Enfin, insérer la tâche de le crontab !

Étape n°3 : installer les outils de publication depuis ce fil RSS vers Diaspora*.

Ici, selon un principe comparable à la publication vers Mastodon, nous allons mettre en place un robot qui publiera automatiquement vers Diaspora* (et en cascade vers Twitter).

Mise en place de Feeddiasp :

- ~\$ sudo pip install feeddiasp
- ~\$ sudo nano /usr/bin/news.py

```
from feeddiasp import FeedDiasp, FBParser, RSSParser

#Synchro des posts depuis un flux RSS
rss = RSSParser(url='URL DU FLUX RSS')
bot = FeedDiasp(parser=rss, pod='URL POD', username='UTILISATEUR DIASPORA', password='MOT DE PASSE',
db='posts.txt')
bot.publish()
```

Enfin, idem, on ajoute une tache cron : ~\$ sudo crontab -e

Et ajouter :

```
* /5 * * * * /usr/bin/python /usr/bin/news.py
```

Mise en place de la publication manuelle Diaspora* vers Twitter

Pour cela rien de plus simple : dans les options de votre compte Diaspora*, il suffit de connecter Diaspora* à votre compte Twitter.

Une fois cela fait, vous pourrez publier un message manuellement sur diaspora* et celui-ci sera rediffusé automatiquement vers Twitter si vous cochez « l'oiseau » !

Étape n°4 : installer les outils de publication depuis ce fil RSS vers Twitter.

L'installation est relative simple et rapide :

- `~$ sudo pip3 install feed2tweet`
- `~$ sudo nano /path/to/feed2tweet.ini` (*remplacer /path/to/ par le chemin vers votre feed2tweet.ini*)

Et compléter de la manière suivante en adaptant avec vos paramètres :

```
[twitter]
consumer_key=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
consumer_secret=bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
access_token=ccccccccccccccccccccccccccccccccccc
access_token_secret=dddddddddddddddddddddddddddddd

[cache]
cachefile=cache.db

[rss]
uri_list=/home/pi/feed2toot/rsslist.txt
tweet={title} {link}

[hashtaglist]
several_words_hashtags_list=severalwordshashtaglist.txt
```

- Ensuite, créer le fichier rsslist.txt qui contiendra la liste des RSS à suivre et à publier automatiquement et y copier-coller vos RSS (*utilise ici la même liste que pour feed2toot, ce qui centralise un peu les réglages*) :
 - `~$ mkdir /home/pi/feed2toot`
 - `~$ nano /home/pi/feed2toot/rsslist.txt`
- Pour tester : `~$ feed2tweet -c /path/to/feed2tweet.ini`
- Pour automatiser, éditer votre cron comme pour Feed2toot et FeedDiasp !

Important : afin d'éviter de polluer votre fil par un tsunami de toots, n'oubliez pas de faire la commande suivante :

```
~$ sudo feed2tweet --populate-cache -c /etc/feed2tweet/feed2tweet.ini
```

Cela permet d'intégrer à la base de données les articles déjà présents dans les flux !

Remarque : par défaut, l'outil de publication feed2tweet tweete en 140 caractères. Pour passer à 250, il suffit d'éditer le fichier de configuration suivant :

```
~$ cd /usr/local/lib/python3.4/dist-packages/feed2tweet
```

```
~$ sudo nano addtags.py
```

En remplaçant la valeur de la ligne 33 « maxlength » de la manière suivante : **maxlength = 250**

Bien entendu, chacun des 3 outils présentés dans cet article pourra être personnalisé pour ajouter des messages spécifiques, publier automatiquement tout RSS d'un site, d'un auteur ou répondant à un mot clé... Bref, c'est hautement personnalisable.

En complément : si vous voulez aller plus loin avec la publication automatique sur Twitter, je vous recommande d'explorer <https://apps.twitter.com> et <https://ifttt.com>.

Remarque non négligeable : la version actuelle (au moment de la rédaction de ce document) de feed2tweet limite la publication aux messages de 140 caractères maximum. Autrement dit, si vous ne modifiez pas le code source de l'outil, tout message basé sur un RSS de plus de 140 caractères ne pourra pas être publié (pour Mastodon et feed2toot, la limitation est à 500 caractères, ce qui laisse bien plus de marge).

Étape n°5 : enjoy !

Petit bonus non négligeable, l'installation du client TinyTinyRSS sur votre appareil mobile vous permettra également, une fois synchronisé à votre serveur d'agrégation RSS, de publier en un clic sur tous vos réseaux, sans passer par un tiers. Elle est pas belle la vie ? Enjoy !

Quels gains au final ?

- ✓ Une fois la phase de configuration terminée : le gain important de temps, la rediffusion se résumant à un clic unique pour l'ensemble de mes réseaux sociaux.
- ✓ Même pendant les vacances, les comptes ne sont pas totalement inactifs, ce qui leur assure une meilleure visibilité.
- ✓ Meilleur respect et meilleur contrôle de ma vie privée en comparaison des outils propriétaires actuellement disponibles (et seuls sont rendus publics les articles de ma veille que je souhaite partager).
- ✓ Contrôle total de l'outil utilisé, et pas l'inverse.
- ✓ Aucune fuite de données involontaires et respect de la vie privée.

Quels points de vigilance ?

- x Temps de mise en place technique : il m'a fallu tout de même consacrer quelques heures de mon temps libre pour mettre en place l'outil, rédiger ce tuto et le partager.
- x Un temps supplémentaire sera ensuite nécessaire pour sélectionner judicieusement la liste des flux pour la rediffusion automatique.
- x Pas simple de faire prendre conscience aux néophytes de l'intérêt de cette démarche.
- x Nécessité de lancer une vraie réflexion quant à l'angle des publications (ce qui peut être également un point positif pour certains).
- x Veillez à ne pas transformer votre bot en machine à spam... Car autant 20 à 30 messages par jour, étalés sur 24h, ça passe, autant 20 messages en 5 minutes feront fuir vos abonnés !

