

---

# Wallabag sur votre Kobo en un clic

Produced by wallabag with tcpdf

Please open [an issue](#) if you have trouble with the display of this E-Book on your device.

---

# Wallabag sur votre Kobo en un clic

[Wallabag](#), ça gère. Je ne m'en passe pas. Mais lire sur une liseuse, c'est quand même beaucoup plus agréable que sur l'écran de l'ordinateur. Alors tous les matins en prenant mon café :

1. Je lançais [calibre](#) pour récupérer le flux RSS de mon instance wallabag et en faire un ebook.
2. Je branchais ma liseuse.
3. J'allumais ma liseuse.
4. J'attendais que calibre fasse le transfère.
5. Je fermait calibre.
6. Je débranchais ma liseuse.
7. Je lisais (*et mon café avait eu le temps de refroidir*)

Qu'on se le dise, six étapes avant de pouvoir lire, c'est long, et le café froid pas bon. Ce matin, je me suis dit qu'il était temps de faire mieux. Et depuis :

1. J'allume ma liseuse.
2. Je clique sur une tuile de l'écran d'accueil.
3. J'attends quelques secondes.
4. Je lis (*et le café est encore chaud :D*)

Le gros gain c'est de ne plus avoir à brancher/débrancher la liseuse et lancer calibre.

Cet article vise donc à présenter la procédure que j'ai suivie pour réaliser cela.

## Construire l'epub chaque matin

Pour cela, j'ai trouvé deux solutions. Dans les deux cas, il vous faut un serveur, ou une machine connectée (*RaspberryPi par exemple*), car l'idée est de lancer un script toutes les nuits pour récupérer vos articles. (*si vous utilisez Wallabag, y'a des chances que vous ayez ça sous la main.*) Et ensuite d'aller le chercher à une URL donnée depuis votre liseuse.

Donc deux solutions, disais-je. Une qui utilise calibre, installé sur le serveur, et l'autre qui utilise la fonction d'export epub de wallabag. Je présente les deux, car je suppose que certains ne veulent pas mettre calibre et toutes les dépendances sur leur serveur juste pour ça. Comme j'ai les deux solutions sous la mains, autant les partager.

### Avec calibre

Quand calibre est installé, il vient avec un programme qui s'appelle *ebook-convert*. À ce programme, on peut lui donner à manger une recette et en échange, il vous donne un epub. Bien pratique. J'ai donc fait un petit script bash qui ressemble à ça :

```
#!/bin/bash
RECIPE="/chemin/vers/wallabag.recipe"
OUTPUT_FILENAME="/un/dossier/accessible/en/ligne/news.kepub.epub"
```

---

```
ebook-convert $RECIPE $OUTPUT_FILENAME --smarten-punctuation \  
--enable-heuristics \  
--pretty-print
```

Où `$RECIPE` pointe vers la recette calibre qui va bien (*on voit ça juste après*) et où `$OUTPUT_FILENAME` pointe vers là où vous souhaitez sauvegarder l'ebook final. Il faut que ce fichier soit accessible en ligne pour qu'on puisse ensuite le récupérer depuis la liseuse. Mettez le dans un dossier avec un nom un peu chelou pour éviter que des gens y aient accès (*comme si le nom du dossier était un token quoi...*). Par exemple, `$OUTPUT_FILENAME` peut ressembler à :

```
OUTPUT_FILENAME="/var/www/mes-super-news-de-wallabag-  
uG89km4TUZ/news.kepub.epub"
```

Et maintenant, la recette calibre. C'est un fichier python qui contient ceci :

```
#!/usr/bin/env python2  
# vim:fileencoding=utf-8  
from __future__ import unicode_literals, division, absolute_import, print_  
function  
from calibre.web.feeds.news import AutomaticNewsRecipe  
class BasicUserRecipe1424519281(AutomaticNewsRecipe):  
    title = 'Wallabag'  
    oldest_article = 7  
    max_articles_per_feed = 100  
    auto_cleanup = True  
    feeds = [  
        ('Wallabag\u2019s articles', 'LIEN_VERS_RSS_WALLABAG'),  
    ]%
```

où vous remplacez `LIEN_VERS_RSS_WALLABAG` par le lien avec le token vers le flux RSS de votre article wallabag (plus d'info [ici](#)). C'est un fichier que j'ai généré avec l'interface graphique de calibre. Toutes les options de configurations sont dans la [doc](#). Vous noterez également, que vous pouvez mettre autant de flux RSS que vous voulez dans la liste `feeds`. Chaque élément étant un tuple (*nom\_du\_flux, url\_du\_flux*).

Et voilà. Vous avez maintenant un script qui télécharge vos articles et les mets dans un fichier epub. Bien.

---

## Sans calibre

Rapidement, comment faire pareil sans calibre ? Et bien avec un petit script python qui va bien. Sauf qu'on ne peut malheureusement pas dire à wallabag : *regarde mon beau token, tu me donnes mon ebook ?*. Non, non, il faut se logger par le formulaire et ensuite récupérer l'epub. Je ne m'étends pas dessus. Le script que je propose est grosso modo un copier/coller de [celui-ci](#) un peu adapté à la situation.

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
import cookielib
import mechanize
WALLABAG_URL = 'https://votre.domaine.fr/wallabag'
OUTPUT_FILE = '/un/dossier/accessible/en/ligne/news.kepub.epub'
LOGIN = 'votre login'
PASSWORD = 'votre mot de passe'
# Browser
br = mechanize.Browser()
# Enable cookie support for urllib2
cookiejar = cookielib.LWPCookieJar()
br.set_cookiejar(cookiejar)
# Browser options
br.set_handle_equiv(True)
br.set_handle_redirect(True)
br.set_handle_referer(True)
br.set_handle_robots(False)
br.set_handle_refresh(mechanize._http.HTTPRefreshProcessor(), max_time=1)
br.addheaders = [('User-agent', 'Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.1) Gecko/2008071615 Fedora/3.0.1-1.fc9 Firefox/3.0.1')]
# authenticate
br.open(WALLABAG_URL)
br.select_form(name="loginform")
br['login'] = LOGIN
br['password'] = PASSWORD
br.submit()
url = br.open(WALLABAG_URL + '/?epub&method=category&value=home')
with open(OUTPUT_FILE, 'w') as newsfobj:
    newsfobj.write(url.read())
```

Vous avez juste à modifier les quatre premières variables et ça devrait fonctionner. Dans le cas où quelque chose ne va pas (*authentification ou autre*), pas de message d'erreurs, juste un ebook avec n'importe quoi dedans.

---

## Récupérer vos articles sur votre Kobo

### La tâche cron

À ce stage, on a un script (*au choix, avec ou sans calibre*) qui peut générer votre ebook et le placer dans un dossier pour qu'il soit accessible en ligne. Il faut donc que ce script soit appelé chaque nuit. Donc une petite tâche cron qui va bien :

```
$ su www-data
$ crontab -e
```

La première ligne sert à s'identifier comme l'utilisateur qui peut écrire dans votre répertoires web (*donc à ajuster selon votre cas*). Et la tâche cron qui va bien :

```
42 4 * * * /chemin/vers/news.sh > /dev/null
```

ou bien vers *news.py* si vous avez opté pour la solution sans calibre. Dans les deux cas, il faut bien sur rendre le script exécutable (`chmod +x news.sh`)

### La config de la Kobo

Encore une fois ici, deux méthodes différentes. Une qui consiste à installer `wget` sur la Kobo pour synchroniser un **dossier** accessible sur internet et votre kobo. L'autre consiste à utiliser le navigateur internet de la Kobo pour récupérer uniquement votre ebook wallabag.

La méthode avec `wget` me semble mieux car elle permet de synchroniser tout un dossier. Donc un ebook à envoyer ? Y'a juste à le mettre dans le dossier qui-va-bien. Mais l'inconvénient, c'est que faut jouer un peu la Kobo... mais bon.

#### Avec `wget`

Pour faire la synchronisation avec `wget`, il faut télécharger ce code : <https://github.com/wernerb/kobo-wget-sync>. Le README est assez clair, mais je vais tout de même relister les étapes à faire ici. Donc une fois le code téléchargé/cloné :

1. On construit le package avec la commande :

```
$ ./package.sh
```

- 
2. Ça va vous faire un fichier build/KoboRoot.tgz. Il faut brancher la Kobo en USB et mettre ce fichier dans le dossier .kobo de votre liseuse. Vous l'éjectez. Elle devrait redémarrer.
  3. Lancez la synchronisation (ça va lancer le script installé et créer les fichiers de config).
  4. On rebranche la Kobo, il y a maintenant un dossier .wget-sync dedans.
  5. Éditer le config.sh. Pour que ça colle bien avec vos paramètres. Vous pouvez aussi mettre une protection via htaccess, donc c'est pas mal.
  6. Quand vous avez fini, vous éjectez la kobo. En cliquant sur le bouton *Sync* de la Kobo, le contenu du dossier devrait être téléchargé. En cas d'erreur, les fichiers logs sont dans le dossier .wget-sync.

Et voilà, maintenant à chaque synchronisation de votre liseuse, le contenu du dossier sera téléchargé. Cool, non ?

Attention, à chaque synchronisation, Kobo récupère tout plein de stats sur vous. Pour plus d'informations et comment limiter un peu la casse [sur cet article](#).

## Sans wget

Sans doute certains n'ont pas envie d'ajouter des programmes sur leur liseuse. Ça tombe bien, voici une autre solution. (*C'est celle que j'utilisais avant*). Donc pour configurer la Kobo pour aller récupérer votre ebook, on utilise le navigateur web intégré. Pour le trouver il faut aller dans *Paramètres > Fonctionnalités Beta > Navigateur internet*. Vous cliquez dessus, il s'ouvre. En tapant l'adresse vers à laquelle est accessible votre ebook dans le barre de recherche, vous devriez pouvoir le télécharger et le lire. Si ça fonctionne, c'est bon, on continue. Sinon, c'est que quelque chose s'est mal passé dans les étapes précédentes. (Je suppose que l'ebook est bien téléchargé.)

Taper l'adresse à la main, c'est un peu long. Donc l'idée est de mettre en favoris cette adresse. Pour la démo, je suppose que l'adresse est <https://mon-wallabag.org/mes-super-news-de-wallabag-uG89km4TUZ/news.kepub.epub>. C'est bien évidemment à adapter à votre cas. On veut faire un favori, sauf qu'on ne peut pas mettre un favori vers un fichier à télécharger (*ça serait trop beau*). Donc il faut faire ça à la main. Pour ce faire, on branche la Kobo. Vous allez ensuite dans le dossier *.kobo/Kobo/* de votre liseuse. Là, il y a un fichier qui s'appelle *Kobo eReader.conf*, vous l'ouvrez. Et vous recherchez la ligne suivante :

```
[ BROWSER_FAVOURITES_GROUP ]
```

Si vous ne la trouvez pas, il faut l'écrire à la fin du fichier. Puis juste sous cette ligne, on met l'adresse de notre ebook. Mais, petite difficulté, il faut l'encoder en "caractères url-safe" 2 fois ! Ça peut se faire sur [cette page](#) par exemple. Ou bien avec [urllib](#). Je répète : il faut le faire deux fois de suite, sinon ça ne marchera pas. Donc dans mon adresse d'exemple ça donne :

---

https%253A%252F%252Fmon-wallabag.org%252Fmes-super-news-de-wallabag-uG89km4TUZ%252Fnews.kepub.epub

Et pour finir, à la fin de cette ligne, il faut ajouter le symbole =. Donc ça donne les deux lignes suivantes à ajouter au fichier de conf :

```
[BROWSER_FAVOURITES_GROUP]
https%253A%252F%252Fmon-wallabag.org%252Fmes-super-news-de-wallabag-uG89km4TUZ%252Fnews.kepub.epub=
```

Et voilà. On a (*presque*) fini. Maintenant, pour télécharger vos articles tous frais, il faut juste ouvrir le navigateur (*qui du coup, se trouvera sur votre écran d'accueil et sera donc facilement accessible*). Puis cliquer sur le bouton en bas à droite du navigateur pour faire apparaître le menu, puis aller dans les favoris, cliquer sur le lien et TADA... ça télécharge.

Cette solution est déjà plus rapide qu'avant, mais on peut faire encore un peu mieux. Surtout que le titre de l'article c'est "*en un clic*", donc on va faire améliorer ça pour faire *un clic* seulement. Personnellement, je ne me sers **jamais** du navigateur de ma liseuse. Je trouve que c'est pas adapté. Donc, pour faire la synchronisation encore plus facilement, j'ai choisi de mettre le lien de téléchargement en page d'accueil du navigateur. Comme ça, dès que je clique sur l'icône *Navigateur internet* de ma liseuse, il me télécharge l'ebook. (*le navigateur n'est plus utilisable pour autre chose... mais comme je ne m'en servais pas... ça me fait une belle jambe.*)

Le problème est le même qu'avant : on ne peut pas mettre un lien de téléchargement en page d'accueil. Donc il faut bidouiller le fichier de config. Et cette fois, il faut rajouter les deux lignes suivantes :

```
[Browser]
homePage=https://mon-wallabag.org/mes-super-news-de-wallabag-uG89km4TUZ/news.kepub.epub
```

Comme ça, tel quel, sans guillemet, sans encodage spécial, sans symbole = à la fin. Et ça devrait tomber en marche tout seul. Vous allez sur l'écran d'accueil de la liseuse, vous cliquez sur *Navigateur internet*, ça vous propose de télécharger le fichier et c'est gagné.

Bonne lecture ;)

---

Published In [informatique](#)

Tags: [koboebook](#)

## blogroll

- [sebsauvage](#)
- [couleur-science](#)
- [sam et max](#)
- [yuntux](#)