

Gestion de documents

- [Espace de stockage : Nextcloud](#)
- [Gestion électronique de documents : Mayan EDMS](#)
- [Inventaire de livres physiques : Inventaire.io](#)
- [Wiki : Bookstack et DokuWiki](#)
- [Gérer sa bibliothèque électronique : Calibre](#)
- [Signature électronique de documents : Documenso](#)

Espace de stockage : Nextcloud

Comme la plupart des gens, j'utilise [Nextcloud](#), avec le client pour pc pour synchroniser des dossiers et le client pour Android pour envoyer automatiquement des fichiers sur le serveur.

Gestion électronique de documents : Mayan EDMS

Bulletins de salaires, dossier scolaire du gamin, factures, modes d'emploi... Oui, tout ça peut se gérer en mettant ça dans un [Nextcloud](#) dans des dossiers et sous-dossiers, mais cela ne permet pas de lier des documents entre eux (genre la facture d'un appareil avec son mode d'emploi) ni de multi-classer un fichier (genre dans un dossier `Assurance` et en même temps dans un dossier `Appartement`).

C'est pour cela que j'ai installé [Mayan EDMS](#) sur mon serveur.

Attention : l'installation en elle-même ne pose pas particulièrement de problèmes mais il n'est pas simple à prendre en main. J'ai acheté le [bouquin](#) et ça m'a permis de mieux appréhender les différents aspects de Mayan. Ce n'était pas du luxe.

Dans les points intéressants du logiciel :

- possibilité de créer des index soit-même sur les critères qu'on veut
- possibilité de mettre des documents dans plusieurs « classeurs »
- possibilité de créer des méta-données sur les documents
- possibilité d'avoir plusieurs versions d'un même document
- recherche de doublons
- possibilité de créer des actions automatiques (genre classement automatique dans un classeur, renommage du document selon ses méta-données, etc)

Ma doc de mise à jour

Adapté de la [doc officielle](#) (lien archive.org car la doc d'upgrade n'est plus sur sur le [site de la doc officielle](#)).

```
export VERSION=4.11.1
```

Récupérer les versions des dépendances (copier / coller le résultat de la commande dans le terminal) :

```
curl -s https://gitlab.com/mayan-edms/mayan-edms/-/raw/series/4.11/config.env |  
grep "^PYTHON" |
```

```
sed -e "s/^/export /"
```

```
supervisorctl stop mayan-edms-celery-beat \  
    mayan-edms-gunicorn \  
    mayan-edms-worker_a \  
    mayan-edms-worker_b \  
    mayan-edms-worker_c \  
    mayan-edms-worker_d \  
    mayan-edms-worker_e &&  
cp /etc/supervisor/conf.d/mayan-edms.conf /etc/supervisor/conf.d/mayan-edms.conf.bak &&  
sudo --user=mayan /opt/mayan-edms/bin/pip install --no-cache-dir \  
    pip==${PYTHON_PIP_VERSION} \  
    setuptools==${PYTHON_SETUPTOOLS_VERSION} \  
    wheel==${PYTHON_WHEEL_VERSION} &&  
sudo --user=mayan /opt/mayan-edms/bin/pip install --no-cache-dir \  
    amqp==${PYTHON_AMQP_VERSION} \  
    mysqlclient==${PYTHON_MYSQL_VERSION} \  
    pycopg==${PYTHON_PSYCOPG_VERSION} \  
    redis==${PYTHON_REDIS_VERSION} &&  
sudo --user=mayan /opt/mayan-edms/bin/pip install mayan-edms==${VERSION} &&  
sudo --user=mayan MAYAN_MEDIA_ROOT=/opt/mayan-edms/media/ \  
    /opt/mayan-edms/bin/mayan-edms.py platforms_template supervisorctl \  
    | sudo tee /etc/supervisor/conf.d/mayan-edms.conf &&  
vimdiff -c 'map <F2> :diffget<cr>]czz | map <F3> ]czz | syn off | windo set wrap | winc h' \  
    /etc/supervisor/conf.d/mayan-edms.conf \  
    /etc/supervisor/conf.d/mayan-edms.conf.bak
```

```
sudo --user=mayan MAYAN_MEDIA_ROOT=/opt/mayan-edms/media/ \  
    /opt/mayan-edms/bin/mayan-edms.py common_perform_upgrade &&  
    supervisorctl start mayan-edms-celery-beat \  
        mayan-edms-gunicorn \  
        mayan-edms-worker_a \  
        mayan-edms-worker_b \  
        mayan-edms-worker_c \  
        mayan-edms-worker_d \  
        mayan-edms-worker_e
```

Inventaire de livres physiques : Inventaire.io

J'utilise [Inventaire](#), directement sur le site officiel. C'est un des rares logiciels que je n'installe pas sur mon serveur, pour plusieurs raisons :

- la complexité (ce n'est pas juste un dossier php + une base de données, c'est un poil plus complexe)
- l'intelligence collective : à l'instar de Wikipédia et de Wikidata (qu'il utilise), les utilisatrices d'Inventaire vont enrichir et améliorer la base de données d'Inventaire et de Wikidata en corrigeant les notices de leurs livres ou en les créant.

Le plus : pouvoir se contenter de scanner le code-barre d'un livre pour l'ajouter à sa collection.

Wiki : Bookstack et DokuWiki

Bookstack

Vous l'avez sous les yeux, j'ai choisi [Bookstack](#) pour mon wiki.

Son organisation en étagères, livres, chapitres, pages me parle énormément, moi qui aime les livres (j'ai même un DUT métiers du livre). C'est une contrainte, mais cela permet aussi de se libérer de la gestion de la structure de son wiki.

En PHP/MySQL, il est simple à installer et extrêmement rapide.

Points intéressants :

- on peut créer des modèles de page
- on peut choisir d'écrire en [Markdown](#)
- on peut ajouter des méta-données qui pourront servir pour la recherche ([exemple](#))
- pense-bête : [la doc de mise à jour](#)

Flux RSS des dernières pages

Documentation de base sur <https://www.bookstackapp.com/hacks/simple-page-rss-feed/>.

Créez un dossier dans `themes`, utilisez ce thème dans le fichier `.env` :

```
APP_THEME=le_nom_du_dossier
```

Créez un fichier `functions.php` :

```
<?php

use BookStack\Entities\Models\Page;
use Illuminate\Support\Facades\Route;

Route::get('/rss/pages/new', function() {
    $pages = Page::query()
        ->visible()
        ->orderBy('created_at', 'desc')
```

```

->take(20)
->get();

return response()->view('rss', ['pages' => $pages], 200, ['Content-Type' => 'text/xml']);
});

```

Créez un fichier `rss.blade.php` (le mien diffère de celui de la documentation), en changeant l'adresse et le titre :

```

<?xml version="1.0"?>
<rss version="2.0"
    xmlns:content="http://purl.org/rss/1.0/modules/content/"
    xmlns:wfw="http://wellformedweb.org/CommentAPI/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
    xmlns:slash="http://purl.org/rss/1.0/modules/slash/"
    >
<channel>
    <title>Derniers articles wiki Fiat Tux</title>
    <link>{{ url('/') }}</link>
    <atom:link href="https://wiki.fiat-tux.fr/rss/pages/new" rel="self"
type="application/rss+xml" />
    <description>Les derniers articles du wiki Fiat Tux</description>
    <lastBuildDate>{{ date(DATE_RSS) }}</lastBuildDate>
    @foreach($pages as $page)
        <item>
            <title>{{ $page->hasChapter() ? $page->chapter->name.' - ' : '' }}{{ $page->name
}}</title>
            <link>{{ $page->getUrl() }}</link>
            <description>{{ $page->getExcerpt() }}</description>
            <content:encoded><![CDATA[{{! isset($page->renderedHTML) ?
preg_replace('/href="\s\/books/', 'href="https://wiki.fiat-tux.fr/books/', $page->renderedHTML)
: preg_replace('/href="\s\/books/', 'href="https://wiki.fiat-tux.fr/books/', $page->html)
!!}}]]></content:encoded>
            <pubDate>{{ $page->created_at->format(DATE_RSS) }}</pubDate>
            <guid>{{ $page->getUrl() }}#{{ $page->id }}</guid>
        </item>
    @endforeach
</channel>

```

```
</rss>
```

Dans le menu Préférences > Personnalisation, ajoutez ceci dans « HTML personnalisé dans l'en-tête », en modifiant l'adresse et le titre :

```
<link rel="alternate" type="application/rss+xml" title="Derniers articles wiki Fiat Tux"
href="https://wiki.fiat-tux.fr/rss/pages/new" />
```

Dokuwiki

Pour le boulot, je choisis plutôt [DokuWiki](#). Il a l'avantage d'être sans base de données, donc simple à sauvegarder et à restaurer.

Austère de base, on peut étendre ses fonctionnalités de façon très agréable avec ses nombreux plugins, comme lister automatiquement les pages d'une catégorie, faire des redirections, ajouter des méta-données qu'on pourra utiliser dans les listes automatiques...

Gérer sa bibliothèque électronique : Calibre

Calibre

[Calibre](#) est un logiciel multi-plateforme qui permet de gérer sa bibliothèque d'epub et autres livres électroniques, de les transférer vers sa liseuse et qui possède bien d'autres atouts (conversion de formats, récupération de flux RSS pour en faire des epub...).

Calibre permet de gérer plusieurs bibliothèques différentes (elles seront dans des dossiers séparés).

NB : si vous voulez accéder à votre bibliothèque calibre via un montage CIFS (Samba), [il faut ajouter l'option `noctrl` aux options de montage](#) sinon vous aurez des problèmes de `database is locked`.

Calibre-web

[Calibre-web](#) permet d'exposer sa bibliothèque calibre au travers d'un site web.

Il permet le téléchargement, la consultation, le téléversement... en mode authentifié ou non, avec la possibilité d'affecter des droits différents selon les utilisateurs.

Très simple à utiliser, j'adore !

Signature électronique de documents : Documenso

J'ai une petite activité d'auto-entrepreneur (quelque soit le nom qui est maintenant donné à ce genre d'activité, j'ai autre chose à faire que d'apprendre les nouvelles désignations par un gouvernement qui n'a que ça à faire que de trouver de nouveaux noms aux choses) et pour éviter à mes clients d'imprimer mon devis, le signer, le scanner et me l'envoyer (ou de s'amuser à mettre eux-mêmes un scan de leur signature sur le pdf), j'ai cherché une solution de signature électronique de documents.

Le grand gagnant est [Documenso](#).

Je l'avais déjà installé et testé, mais j'avais attendu avec impatience la compatibilité avec Nextcloud 29 de [Libresign](#), et cela semblait fonctionner (d'après mes tests). L'avantage aurait bien évidemment été l'intégration à Nextcloud, plus besoin de téléverser le fichier à signer, hop, pratique.

Mais un client n'a pas réussi à signer un document. Comme j'ai autre chose à faire que déboguer ça avec un client à l'autre bout du fil, j'ai décidé de rester sur Documenso.

Installation

Bien évidemment, comme la plupart des logiciels de nos jours, la méthode d'installation suggérée est Docker. Pouah.

Création d'un utilisateur dédié

```
adduser --disabled-login \  
  --disabled-password \  
  --home /var/www/documenso \  
  --no-create-home \  
  --shell /bin/false \  
documenso
```

Création d'une base de données PostgreSQL

```
sudo -u postgres createuser -P documenso
sudo -u postgres createdb -0 documenso documenso
```

Récupération du code

```
export VERSION=1.11.0
```

```
cd /var/www
git clone https://github.com/documenso/documenso.git
cd documenso
chown -R documenso: .
sudo -u documenso git checkout -b "v$VERSION" "v$VERSION"
```

Configuration

```
sudo -u documenso cp .env.example .env
sudo -u documenso vimdiff -c 'syn off | windo set wrap | winc h' .env .env.example
```

NB : le `vimdiff` ne sert que lors d'une mise à jour, lors de l'installation initiale, contentez-vous d'éditer le fichier

Compilation

```
sudo -u documenso npm ci
sudo -u documenso npm run build
```

Migration de la base de données

```
sudo -u documenso npm run prisma:migrate-deploy
```

Service systemd

```
cat <<EOF > /etc/systemd/system/documenso.service
[Unit]
Description=documenso
After=network.target
After=postgresql.service
```

```
Requires=postgresql.service
PartOf=postgresql.service

[Service]
Type=simple
User=documenso
Environment=PATH=/var/www/documenso/node_modules/.bin:/usr/local/bin:/usr/bin:/bin
WorkingDirectory=/var/www/documenso
ExecStart=/usr/bin/npm run start
TimeoutSec=15
Restart=always
SyslogIdentifier=documenso

[Install]
WantedBy=multi-user.target
EOF
systemctl daemon-reload
systemctl enable --now documenso
```

Configuration Nginx

```
location / {
    include /etc/nginx/proxy_params;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_pass http://127.0.0.1:3001;
    proxy_redirect http://127.0.0.1:3001 https://documenso.example.org;
}
```