

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="\Vbooks/',
'href="https://wiki.fiat-tux.fr/books/', $page->renderedHTML) : preg_replace('/href="\Vbooks/',
'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...

Révision #5

Créé 11 mai 2022 09:30:48 par Luc

Mis à jour 12 novembre 2023 12:32:02 par Luc