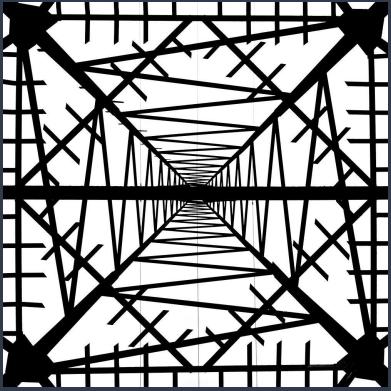


Sulu : LE CMS Ultime?

Meetup AFUP Montpellier

Julien Vinber - 24 mars 2022 - Crealead



Julien Vinber

- Junior depuis 20 ans
- Architecte technique chez CGI
- Coorganisateur des Meetup
- ...
- ...

LinkedIn : [@julienvinber](https://www.linkedin.com/in/julienvinber)

C'est quoi un CMS?



CMS : Content Management System ou Système de gestion de contenu

Le CMS a pour but d'offrir un ensemble d'outils et de mécanisme pour gérer le contenu d'un site internet.

C'est quoi un CMS?

Alors pourquoi tous les projets n'utilisent pas de CMS ?



La simplicité n'est pas un but, mais une ressource.

Alors pourquoi tous les projets n'utilisent pas de CMS ?

Pour nous les dev quand on pense CMS les concepts que l'on va associer :

- Code de merde (non je ne parlerais pas de wordpress)
- Extensible, mais après avoir compris la logique non documentée.
- Absence de liberté.
- Mécanisme absolument incompréhensible pour faire des choses simples dans la vraie vie.
- Il existe un plug-in pour cela (lourd, bugger et ne fait pas vraiment ce que l'on veut.)

Alors pourquoi tous les projets n'utilisent pas de CMS ?

Les CMS sont orienté Webmaster.

Quelquefois ils pensent aux utilisateurs.

Généralement ils concentrent les efforts en documentation à destination des décideurs.

Et le dev dans tout cela?

Alors pourquoi tous les projets n'utilisent pas de CMS ?

Alors pourquoi parler de Sulu?



SULU

=

Symfony (brut)

+

Bundle CMS

Alors pourquoi parler de Sulu?

Si je veux :

- Je peux créer mon entité Doctrine
- Je peux créer mon controller
- Les templates c'est du twig que je peux autant utiliser pour faire mes pages Sulu que pour mes controller.
- ...

Alors pourquoi parler de Sulu?

Donc le CMS Ultime?

Forcément non.

Mais pas dénué d'intérêt,
si on accepte les défauts.

Donc le CMS Ultime?

Avantages



- La promesse et tenus, nous avons la liberté de faire ce que l'on veut en Symfony en oubliant Sulu
- Sulu est souple et permet facilement de personnaliser pour avoir le comportement attendu.
Exemple :
 - Nous avons pu créer une entité et facilement lui attacher tous les comportement des pages normale, telle que la prévisualisation.

Avantages

Fonctionnalités :

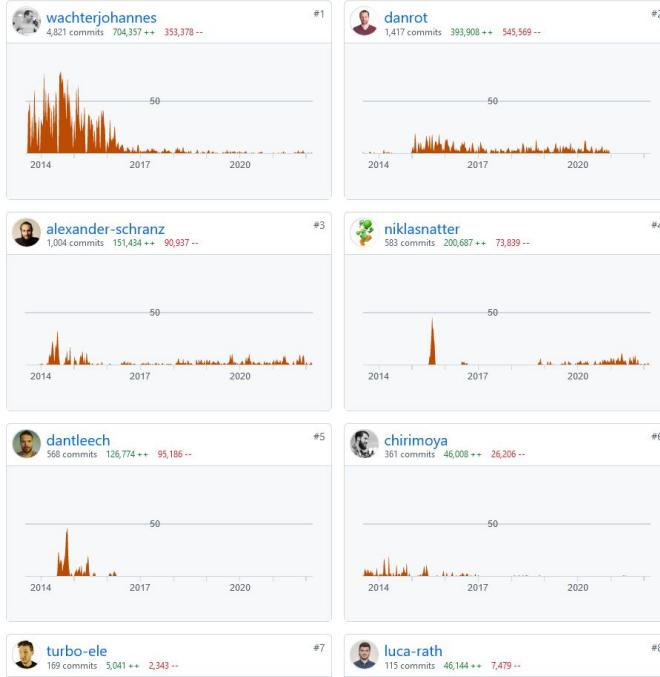
- Gestion de page et leur contenu.
- Pas de système de gros Wysiwyg, mais des formulaires avec la possibilité de créer des champs multiple en laissant l'utilisateur choisir.
- Gestion de média (image / vidéo / document)
- Traitement à la volée sur les images.
- Personnalisation de l'admin.
- Liberté entière pour gérer l'affichage.
- L'admin et le site web sont deux projets distincte.
- Gestion du cache en natif.
- Gestion d'utilisateur et des droits, multilangue, multisite...
- ...

Avantages

Oui, mais...



Le plus gros défaut à mon sens et la taille de l'équipe qui travail sur le projet.



Inconvénient

- Parfois verbeux.
- Documentation bien faite, quand elle aborde le sujet. Mais il manque pas mal de points un peu plus poussé.
- Si on veut vraiment faire ce que l'on veut pour l'admin il vaut mieux bien connaître React.

Inconvénient

Coté perf c'est mitigé.

- C'est un peu lourd
- Entre 200 et 300 requête par page sur un site "moyen riche"
- Sur Windows un site SF c'est un peu lent, mais cela reste utilisable. Avec Sulu c'est quasiment insupportable.
- Une admin en react c'est plutôt lourd pour les phases de dev.
- Une temps moyen d'une petit second par pages

Mais il y a du cache (attention le cache peut se désactiver, par exemple avec un formulaire)

Inconvénient

Beaucoup de micro frustration :

- Une page Sulu, à une route Sulu. Mais la route n'est pas SF. Donc impossible de mélanger.
- Ils utilisent le XML pour décrire les modèle de pages. Mais c'est très lourd et on à exploser les limite de mémoire sur le projet (riche à ce niveaux).
- On peut personnaliser beaucoup de choses dans l'admin, sauf que Sulu charge ses blocs après nous, impossible de personnaliser ce qu'il fond.
- Sulu utilise PHP CR, et passe par doctrine résultat cela fonctionne comme doctrine, mais sans jamais être dû doctrine. Par exemple il y a deux système de fixture, identique, mais différent.

Inconvénient

Des choix discutables

- il y a 2 sous projets, donc 2 gestion de JS. Sauf que l'un est avec NPM, l'autre avec YARN.
- Les utilisateurs cela ce nomme des contacts.
- Les articles c'est avec elasticsearch. (une version sans arriveras peut-être en v3)

Inconvénient

Niveaux techniques

- On peut faire du PHP 8.1, mais rien de planifié pour du SF 6
- Pour l'admin utilise des script custom et lourd pour gérer l'API. On ne peux pas utiliser le sérialiseur de SF ou API platforme pour avoir quelque chose de propre.
 - On c'est retrouvé à exploser la mémoire car leur système aller chercher ultra loin dans les data.
- Ils utilisent de vieux bundle swiftmailer jms serializer

Inconvénient

En conclusion



Non, ce n'est pas le CMS Ultime.

Mais il n'est pas pour autant à jeter.

Si on le connaît, parfait pour de petit site ou il y a un besoin d'avoir la main.

Pour des sites moyens? Je dirait plutôt oui pour des sites qui vont vivre dans le temps avec la même équipe, mais à bien prendre en compte 2 points :

- Le coûts de familiarisation n'est pas négligeables.
- Il n'y a pas "le plug-in qui le fait déjà".
- C'est une solution faite pour les dev. Mais pas sans quelque frustration.

Pour des gros site? Je serais moins chaux, on doit faire des compromis et j'ai peur que leur coût ne soit pas compatible avec la taille du projet.

Pour qui?

Je le réutiliserais?



Ce n'est pas une solution cout de coeur.

Mais je ne l'exclut pas.

?

D'autre CMS à base de SF full stack

<https://symfony.com/projects/contao>

<https://symfony.com/projects/kunstmaanbundles>

<https://symfony.com/projects/bolt>

<https://symfony.com/projects/symfonycmf>