

# Outils pour sites statiques modernes à l'intention du développeur paresseux

David Sferruzza

# Bonjour à tous !

Je m'appelle **David**, je **développe** des sites web (depuis une quinzaine d'années) et je suis **paresseux** (~~depuis~~ toujours).

# Des *sites web* ?

La terminologie est variée :

- site web
- applica on web
- services web
- site Internet



Il n'y a pas de définitions formelles, c'est par pour les *abus de langage* !

# Le principe du site web

On a un **logiciel** qui s'exécute sur un **serveur** pour délivrer du **contenu** grâce au **protocole HTTP** d'un bout à l'autre du **réseau Internet**.

Les **navigateurs** web permettent d'appeler ces contenus, de les **interpréter** et de les **afficher**.

**Sites web : deux familles,  
deux ambiances**

# 1/ Sites statiques

On utilise un logiciel de serveur web **standard** (Apache, nginx, ...) qui va servir du contenu **inerte**.

*Le contenu a été écrit préalablement.*

## 2/ Sites dynamiques

On **développe** une application qui va recevoir chaque requête HTTP et **générer** une réponse potentiellement **personnalisée**.

# Dynamisme 🤞

Le principe des sites dynamiques est beaucoup plus **puissant** !

- on peut faire **pareil ou plus** que les sites statiques
- on peut implémenter des comportements **arbitraires**



# Dynamisme

Sauf que tout ça a un coût en :

- complexité du système
- dépendances à d'autres logiciels
- intégrité des données
- étendue de la surface d'a. aque
- performances
- ...

# Dynamisme

Heureusement, on peut affronter ces problèmes grâce à :

- la recherche scientifique
- des innovations industrielles
- des professionnels du logiciel
- des outils libres

# Content Management System

Un CMS est un site dynamique **pré-développé** pour gérer du contenu *sans avoir besoin de connaissances techniques*.

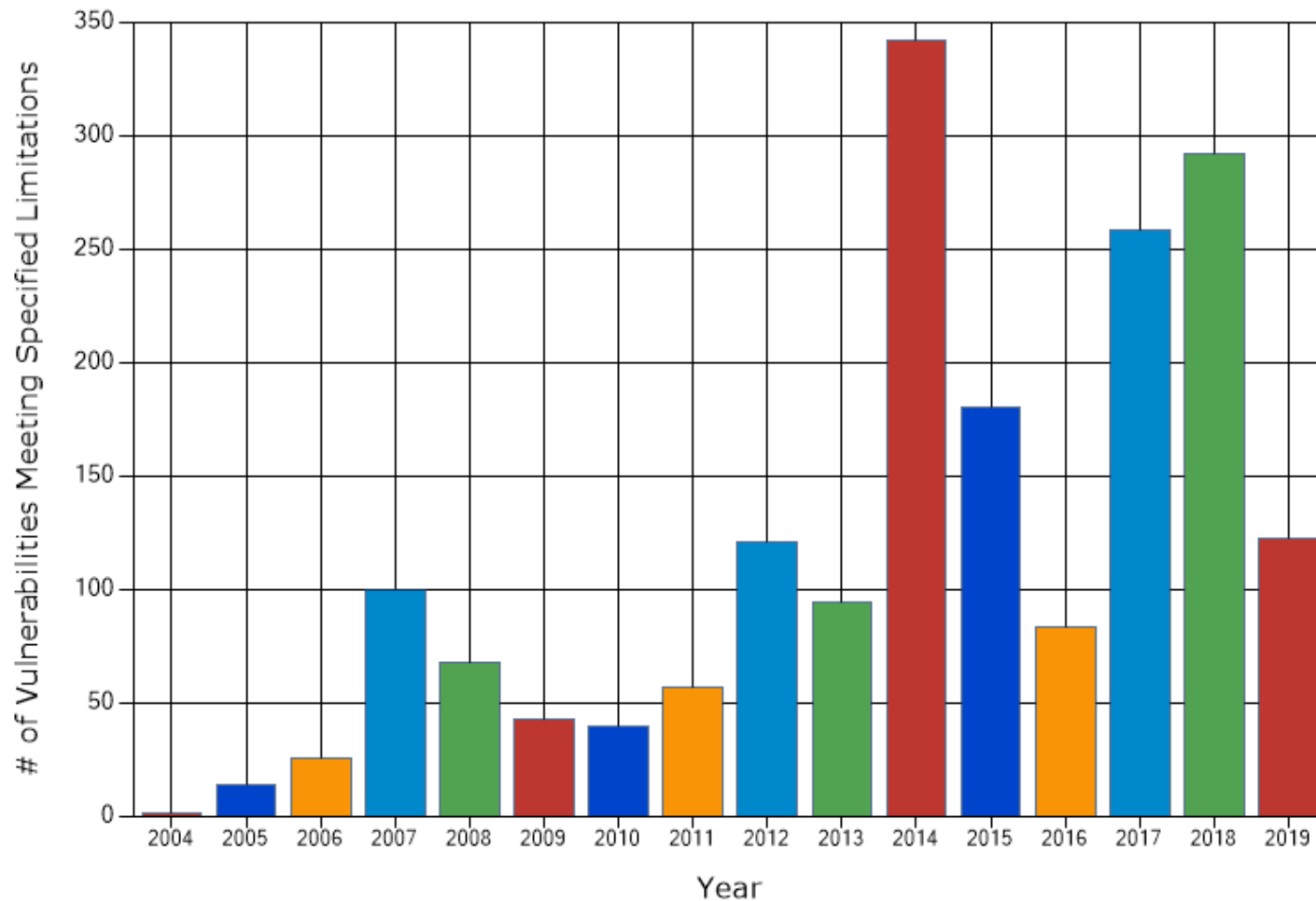
*Exemples : WordPress, Drupal, Joomla!,  
PrestaShop, ...*

Quelques éléments de différenciation :

- hypothèses sur le contenu
- interface d'édition du contenu
- extensibilité
- écosystème (thèmes, *add-ons*)

# CVE liées à WordPress

Total Matches By Year



**Prenons du recul**

# Tout est compromis

En général, quand on fait un site web, le problème n'est pas de trouver une solution qui marche mais **d'évaluer les risques/opportunités** des solutions qui marchent pour choisir la meilleure.

# En réalité

C'est vraiment très difficile. Il y a énormément de paramètres.

→ On choisit souvent à l'instinct 😊

# *Plot twist*

Limitations :

- les biais cognitifs
- l'évolution de l'écosystème
- les connaissances imparfaites

→ **N'arrêtez jamais d'apprendre** 



# Sites dynamiques

Parfois, c'est cool de faire un site dynamique :

- les cas d'usage l'imposent
- accès à des abstractions
- ...

Parfois, c'est cool d'utiliser un CMS :

- pas besoin de réinventer la roue
- pas besoin de développeur
- on a l'habitude
- ...

*Aujourd'hui on va parler des autres cas 😊*

# Le projet du jour

Parfois, on veut un site :

- au contenu gérable par des **non-développeurs**
- pour délivrer du contenu (**peu d'interactions**)
- au rendu **spécifique et performant**
- **facile à maintenir** sur le long terme

# Bienvenue dans ma conférence !

*Avec les bonnes techniques, on peut  
sa sfaire ces contraintes avec un site  
sta que !*

# Approche naïve

On édite et déploie manuellement des fichiers HTML, CSS, JS, ...

*On utilise Git pour versionner tout ça 🙏*

- ✘ travail répétitif, pas d'abstractions/d'outils
- ✘ déploiement sujet à erreurs humaines

# Générateur de sites statiques

On ajoute une étape de construction pour séparer :

- ce que le développeur écrit
- le produit fini (un site statique)

Ça permet d'utiliser des **abstractions** et des fonctionnalités pré-développées.

- langage de gabarit (*templating*)
- pagination
- blog (et flux RSS)
- ...

# Choisir un générateur

Il en existe **plein**. Quelques critères :



- usage principal, fonctionnalités
- format d'entrée
- extensibilité ( $\Rightarrow$  langage utilisé)
- écosystème (modules, documentation, ...)
- facilité d'installation

*Exemples : Jekyll, Hugo, Gatsby, ...*

# Avec un générateur

Une fois la techno choisie et maîtrisée, le développement du site est plus rapide et sa maintenance bien plus facile !

*Comme on a déjà payé le prix d'avoir une étape de construction, on peut l'exploiter encore plus (webpack, ...).*

-  travail répété, pas d'abstractions/d'ouls
-  déploiement sujet à erreurs humaines

# Intégration continue

*Un système lance des tâches à chaque fois qu'une nouvelle version du code est publiée.*

- **générer** (produire des artefacts)
- **vérifier** (éviter le "*ça marche sur ma machine*", lancer les tests automatisés)
- **déployer**



Pour un site statique, on veut surtout générer le site et le déployer.



# GitLab CI

C'est le système d'intégration continue lié à GitLab. Il est top 🙌

```
variables:
  DEBIAN_FRONTEND: noninteractive
  NODE_VERSION: 12

build:
  stage: build
  image: ruby:2.6-stretch
  script:
    - curl -sL https://deb.nodesource.com/setup_${NODE_VERSION}.x
    - apt-get install -y nodejs
    - npm install
    - npm run build
    - bundle install
    - bundle exec jekyll build
  artifacts:
    paths:
      - _site/
  tags:
    - docker
```

# Avec un générateur + CI

- travail répétitif, pas d'abstractions/d'ouïssances
- déploiement sujet à erreurs humaines
- **facile à maintenir** sur le long terme
- contenu gérable par des **non-développeurs**

# Gestion du contenu

Dans la plupart des générateurs, le contenu est à écrire en **Markdown** et il faut savoir manipuler **Git**.

→ C'est un frein pour les non-développeurs 😞

# Gestion du contenu

Heureusement, il existe des **services** d'interface graphique d'édition :

1. On configure le *schéma* des données
2. On saisit les contenus via une interface graphique
3. À chaque génération du site, on récupère les contenus via une API

*Exemples : Conten. ul, Prismic, Strapi, ...*

# Contentful

The screenshot shows the Contentful website with a navigation bar containing: contentful logo, Product, Solutions, Customers, Learn, Pricing, Talk to us, and Try for free. The main heading is "How enterprises deliver better digital experiences". Below it is a paragraph: "You don't need another CMS. You need a better way to manage content – unified, structured, and ready to deploy to any digital channel. Finally, something editors and developers can agree on." Two buttons are present: "Request demo" and "Start building free".

The mobile app interface for Glossier features:

- Brand Logo: **Glossier.**
- Hero Left Model: Image of a woman's face.
- Hero Right Product: Image of three Glossier skincare products.
- Homepage Text: A text block at the bottom.

On the right, a diagram illustrates the Contentful architecture layers:

- Presentation Layer
- Content Model
- JSON Data

# Contentful : schéma



**Post**


Un article de blog [Edit](#)

Fields (9)

JSON preview

⋮	<b>Title</b> Short text	Entry title	Settings	...
⋮	<b>Slug</b> Short text		Settings	...
⋮	<b>Excerpt</b> Long text		Settings	...
⋮	<b>Cover</b> Media		Settings	...
⋮	<b>Content</b> Long text		Settings	...
⋮	<b>Description (meta)</b> Long text		Settings	...
⋮	<b>Published At</b> Date & time		Settings	...
⋮	<b>og:image</b> Media		Settings	...
⋮	<b>Tags</b> References, many		Settings	...

# Contentful : édition

<  Drame : Incapable de scaler son offre, une startup... ?

Title (required)  
Drame : Incapable de scaler son offre, une startup rétrogradée au rang de "PME"  
79 characters Requires less than 256 characters

Slug (required)  
drame-incapable-de-scaler-son-offre-une-startup-retrogradee-au-rang-de-pme

Excerpt (required)

EditorPreview

H B I “ ☰ ☰ 🔗 ⋮ Insert media ▾ ✕

Après deux ans d'existence, la startup "VaCATair" doit désormais se présenter comme une PME et non plus comme une startup. Son dirigeant redoute les futurs impacts business. Témoignage.

M ↓ Format your text like a pro with the [markdown cheatsheet](#).

28 words, 186 characters Requires less than 500 characters

# contentful2md



```
#!/usr/bin/env sh
```

```
export CONTENTFUL_SPACE="..."  
export CONTENTFUL_ACCESS_TOKEN="..."  
export CONTENTFUL_PREVIEW="true"  
export CONTENTFUL_CONTENT_TYPE_ID="post"  
export CONTENTFUL_LOCALE="fr-FR"  
export OUTPUT_DIR="src/_articles"
```

```
npm run contentful2md
```



# Naddymess

**NaddyMess**  
L'actualité startup et innovation  
disruptée avec bienveillance.



il y a 1 an

## Il demande à un freelance de développer le "Facebook des propriétaires canins" pour 500€



La détermination et un budget modeste sont-ils suffisants pour porter une idée brillante ? Retour sur l'histoire de DoggyDog, le réseau social pour propriétaires de chiens lancé il y a quelques mois.

il y a 1 an

## Scandale : Sa Keynote était en fait un PowerPoint



Alors qu'elle présentait sa Keynote depuis 15 minutes, l'auditoire de Marie découvrit avec effroi qu'il s'agissait en fait d'une présentation PowerPoint. En cause : la présence d'une transition "damier".

## Newsletter

Vous avez déjà donné votre adresse email à des gens moins scrupuleux

#WeArePrivacyByDesign

Email

INSCRIPTION

## Fail Tech

Ce site est labellisé Fail Tech



# Écrire un article Naddymess

1. Avoir une idée ~~stupide~~ disruptive
2. Écrire l'article dans Contentful
3. Créer un *pipeline* dans GitLab
4. A. endre ~1 min

# Conclusion

Besoin :

- site web avec **peu d'interactions**
- 1+ développeur à disposition
- (contenu gérable par des **non-développeurs**)

# Conclusion

Bilan (site statique) :

- **facilité de développement**  
(versionnement, abstractions, outils)
- **facilité d'édition**  
(admin UI, CI/CD, multiples sources de données)
- **avantages des sites statiques**  
(sécurité, performances, simplicité)

→ Je peux maintenir une dizaine de sites sans effort 😎

# NetlifyCMS

1. L'admin est **embarquée** dans le site statique
2. Les modifications sont **persistées** grâce à l'API de GitHub/GitLab/Bitbucket
3. Chaque modification entraîne un **re-déploiement** du site

# NetlifyCMS

Bilan :

- potentiel : 😄
- mise en place : 😞
- utilisation : 😐 😊

→ Tout n'est pas fluide mais ça évolue, à surveiller !

# Questions ?



Twitter: [@d\\_sferruzza](https://twitter.com/d_sferruzza)

Slides sur GitHub :

[dsferruzza/conf-sites-staques-modernes](https://github.com/dsferruzza/conf-sites-staques-modernes)