

Beste de savoir

[Chronique]Zest of Dev 5

15 janvier 2019

Table des matières

1.	Introduction	1
2.	Manifeste de la publication partielle	1
3.	Les PR de la quinzaine, une feature est à l'intérieur	3
3.1.	Les PR de zds-site	3
3.2.	ZMarkdown	3
3.3.	Latex Template	4
3.4.	Notable facts	4

% [CHRONIQUE]ZEST OF DEV 5 % artragis % 11 février 2018

1. Introduction

Hey mais ça fait deux semaines que j'ai pas écrit un billet moi. Je vais donc y remédier. Il faut dire qu'on a encore travaillé sur zds. Cette version 27 va vraiment être incroyable.

2. Manifeste de la publication partielle

Pour commencer cette chronique, je vous propose de faire un retour en arrière, dans ce passé lointain où coldplay était au top 50 pour son titre "A sky full of stars".

ÉLÉMENT EXTERNE (VIDEO) —

Consultez cet élément à l'adresse <https://www.youtube.com/embed/VPRjCeoBqrI>.

C'est en août 2014 que nous sommes donc, et zeste de savoir, vient de passer son premier mois en ligne.

Les premiers tutos sont importés du site du zéro mais déjà quelques auteurs s'affairent à rédiger de nouveaux contenus qui émoustilleront vos méninges. Et certains sont chafouins : en passant du site du zéro à zds une fonctionnalité a été perdue : pouvoir publier une partie seulement du tutoriel.

2. Manifeste de la publication partielle

Chez les développeurs, on avait déjà quelques idées de comment implémenter ça, mais d'autres priorités nous attendaient. Alors le [1334ème ticket](#) de notre bugtracker a commencé à prendre la poussière.

Et puis, en 2015 pierre-24 et moi nous sommes lancé dans une aventure assez complexe : refondre complètement le module de tutoriel pour qu'il soit plus facilement évolutif et que le code soit plus lisible. On avait appelé ça la zep 12.

De 9 mois de développements sortiront une conclusion : si on voulait sortir la version 12 du site un jour, il fallait arrêter de vouloir ajouter tout de suite les fonctionnalités qu'il était plus difficile d'ajouter dans le précédent modèle et la publication partielle devint un fossile.

Pendant ce temps, un autre soucis était en train de poindre pour zeste de savoir : python 2 était en fin de vie, et django avait décidé qu'après la version 1.11 la compatibilité avec python 2 ne serait plus assurée. Alors il fallait passer à python3 et ce n'est pas une mince affaire.

Il y a eu plusieurs essais infructueux, mais voilà pierre-24 s'est motivé à reprendre le travail à ce propos et de le mener à bien (et ça a marché).

Du coup, j'ai fait une promesse : si on arrivait avec python 3, alors j'implémenterai la publication partielle. Et c'est en passe d'être mis dans la branche /dev (il manque juste quelques détail de style de code dans le javascript).



Comment faire du coup ?

Il faut comprendre que les contenus (article, tutoriel et billet) sont issus d'un code commun qui sépare la données en deux grands ensembles :

- le contexte du tutoriel : il est stocké en base de donnée. Il rassemble les auteurs, les dates de création, les différentes versions (béta, publique, brouillon), et le classement dans nos catégories ;
- le texte du tutoriel et son organisation : il est stocké dans git chaque texte a son propre fichier et pour assurer la cohérence du tout (ordre des parties, accessibilité des urls...) on a le **manifest.json**.

C'est à ce fichier qu'une fonctionnalité – plus précisément un marqueur– a été ajouté : lorsqu'on écrit un chapitre ou une partie, on peut la marquer comme étant prête (ou non) à être publiée et cela se traduit dans le manifeste par un simple *booléen*.

Par exemple, dans mon tutoriel sur ASP.NET, il y a une partie que j'ai commencée mais que j'aimerais finir avant qu'elle soit publiée mais sans avoir à retarder les petites publications pour cause de fautes d'orthographe, je peux faire :

```
1 {
2   "object": "container",
3   "slug": "creez-votre-site-web-avec-asp-net",
4   "title": "Créez votre site web avec ASP.NET",
5   "introduction": "introduction.md",
6   "conclusion": "conclusion.md",
```

3. Les PR de la quinzaine, une feature est à l'intérieur

```
7     "children": [  
8         // ... (il y a plein de parties)  
9         {  
10            "ready_to_publish": false,  
11            "object": "container",  
12            "slug": "allons-plus-loin-avec-asp-net",  
13            "title": "Allons plus loin avec ASP.NET",  
14            "introduction":  
15                "allons-plus-loin-avec-asp-net/introduction.md",  
16            "conclusion":  
17                "allons-plus-loin-avec-asp-net/conclusion.md",  
18            "children": [  
19                // plusieurs chapitres en rédaction  
20            ]  
21        },  
22    ],  
23    "version": 2,  
24    "description":  
25        "La programmation web en C# accessible à tous !",  
26    "type": "TUTORIAL",  
27    "licence": "CC BY"  
28 }
```

Cette fonctionnalité a été testée avec le système de test frontend qui fait que lorsqu'on met à jour le code de zestedesavoir, automatiquement un test est lancé qui vérifie que la réaction de l'IHM est la bonne. Pour l'instant, je ne teste qu'avec la version bureau du site. On verra plus tard pour ajouter les petits écrans.

3. Les PR de la quinzaine, une feature est à l'intérieur

3.1. Les PR de zds-site

- Un peu de nettoyage des logs sur travis [↗](#) (vhf)
- Mise à jour de messages lors de la rédaction [↗](#) (Anto)
- Un refactoring du code du forum pour avoir des noms cohérents [↗](#) (vhf)
- Corrige la position du texte ajouté lorsqu'on appuie sur citer [↗](#) (rezemika)
- Fix les alertes persistante lorsqu'un contenu est dépublié [↗](#) (artragsis)
- Barre de navigation pour parcourir les tribunes de haut en bas [↗](#) (Anto)
- Tri des sous-catégories [↗](#) (anto)
- Retire les sous-catégories vides [↗](#) (anto)
- Ajout d'un espace insécable entre titre et soustitre [↗](#) (rezemika, les petits pixels)

3.2. ZMarkdown

- Ajout des blocs avec titre et de type neutre [↗](#) (artragsis)
- Améliore et corrige les blocs avec titre et type neutre [↗](#) (vhf)
- Améliore la documentation [↗](#) (ache)

3. Les PR de la quinzaine, une feature est à l'intérieur

- Corrige le latex généré pour insérer des images qui ont des `.` dans leurs noms ↗ (vhf)

3.3. Latex Template

- Ajout du support des blocs neutres ↗ (Héziode)

3.4. Notable facts

Eskimon a décidé que ZMarkdown pouvait être un excellent outil pour créer un blog avec Pélican. Qui sait ? peut être qu'il écrira un billet à ce propos.

Ache a commencé à écrire du code pour générer des QCM/QCU sur zmarkdown.