

*Beste de savoir*

[Chronique] Zest of Dev 3

---

15 janvier 2019



# Table des matières

1.	Introduction . . . . .	1
2.	Zmarkdown et la petite sirène . . . . .	1
3.	Publication PDF, partielle et mesurée . . . . .	2
4.	Les PR de la quinzaine . . . . .	3
4.1.	ZMD : . . . . .	3
4.2.	Zds site . . . . .	3
4.3.	Latex-template . . . . .	4

% [CHRONIQUE] ZEST OF DEV 3 % artragis, victor % 14 janvier 2018

## 1. Introduction

Première chronique de 2018, que s'est-il passé cette semaine sur le développement de zds ? Sans plus attendre voici l'état des lieux.

## 2. Zmarkdown et la petite sirène

Le développement de [zmarkdown](#) [↗](#) a repris et plusieurs modifications ont été apportées :

- victor nous a proposé de retirer le code du "serveur zmarkdown" de la partie zds-site (backend Django de zds) car ce n'est pas au projet zds-site de faire tourner ce serveur. Du coup aujourd'hui le serveur zmarkdown, qui reçoit du Markdown et retourne de l'HTML ou du Latex, est dans le paquet npm `zmarkdown`;
- Héziode a fait une PR pour s'assurer que la syntaxe personnalisée pour les codes *inline* (c'est à dire les codes insérés dans le texte, `comme celui ci`) soit intégrée à notre utilisation de rebber (le convertisseur "arbre syntaxique markdown" -> code  $\LaTeX$ );
- victor a commencé à intégrer mermaid, un outil qui vous permettra d'intégrer des diagrammes à vos tutoriels simplement en écrivant du texte<sup>1</sup>. Une étape ultérieure pourrait être d'intégrer `dot` [↗](#) (graphviz), si cette syntaxe intéresse notre public.

Pour rappel [ZMarkdown](#) [↗](#) est un projet que nous avons initié en 2017 pour remplacer notre interpréteur de markdown (Python-ZMarkdown) que nous avons du mal à maintenir et qui manquait de souplesse, notamment lorsqu'il n'était capable que de transformer du Markdown en HTML, nous forçant à utiliser une solution tierce et particulièrement pénible pour transformer notre code markdown en latex puis en PDF.

---

1. Pour l'utiliser de plus en plus souvent, c'est vraiment bien fichu comme truc et pas trop complexe. Par contre faudra peut être prévoir, un jour, un petit bouton dans notre éditeur pour vous aider à créer vos diagrammes.

### 3. Publication PDF, partielle et mesurée

L'idée est de passer par un écosystème open source – nommé [remark](#) – constitué de bibliothèque permettant de manipuler le markdown sous forme d'un [AST](#) Markdown, sur lequel on peut intervenir avant, pendant, après le *parsing*.

Parmi les possibilités offertes, une transformation indispensable pour zds est celle qui consiste à transformer le [MDAST](#) (AST Markdown) en [HAST](#) (AST HTML), pour enfin projeter l'arbre HAST en HTML pour afficher vos messages et tutoriels.

Nous avons également implémenté une projection vers LaTeX de l'AST Markdown, projet que nous avons nommé [rebber](#).

Afin de structurer ce projet proprement et de permettre à tout le monde de bénéficier de notre travail, nous avons créé un module séparé et indépendant pour chacun des éléments de syntaxe Markdown propres à zds. Ces extensions sont open sources, disponibles sur NPM et [référéncées](#) dans le projet remark.

Vous pouvez les voir [ici](#).

Lorsqu'une extension commence par `remark-` cela veut dire que c'est une extension au parser markdown. Lorsqu'elle commence par `rehype-`, elle concerne la transformation en HTML et, enfin, si cela commence par `typographic-` c'est une extension d'une bibliothèque de typographie nommée [textr](#).

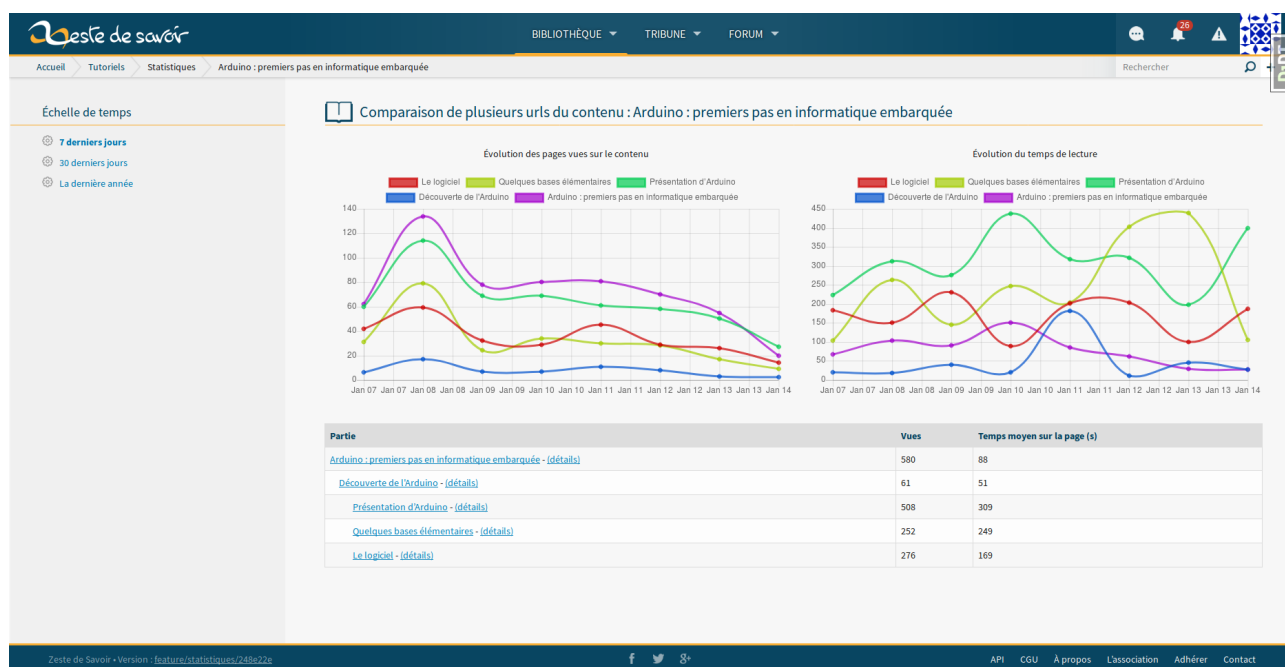
### 3. Publication PDF, partielle et mesurée

Nous avançons de plus en plus en ce qui concerne la publication des PDF.

Bien que la récente refonte des rôles entre les projet zmarkdown et zds-site a fait que nous devons clarifier qui déploie quoi (mais c'est un détail technique, autant pour nous que pour vous) nous avons désormais une pipeline de publication qui fonctionne et les premiers exemples tombent.

Autre grand chantier fonctionnel du site : les statistiques pour les auteurs. Eskimon nous propose un petit aperçu avec des vraies données (bon c'est avec son propre tuto – qui est le plus vu du site – qu'il a joué, mais on le lui pardonnera volontiers, non ?).

## 4. Les PR de la quinzaine



Vous pouvez suivre le travail d'Eskimon et Anto [sur le forum](#) . Je vous encourage à y laisser quelques pouces verts pour encouragement notre caribou national!

Dernière fonctionnalité qui avance : la publication partielle. Lorsque vous rédigez votre tutoriel, vous pouvez dire "cette partie n'est pas prête à être publiée" et dès lors, lorsque le validateur passera et validera votre tutoriel, celle-ci n'apparaîtra pas. Néanmoins, cela vous permettra de continuer à rédiger votre tutoriel et à prévoir son organisation.

Notons que pour l'instant la publication partielle ne concerne pas les minitutos ni les articles.

Le statut de cette fonctionnalité est "Chez moi ça marche". J'ai écrit un test qui simule la navigation, mais celui-ci échoue lorsqu'il est exécuté par travis alors qu'il réussit chez moi.

## 4. Les PR de la quinzaine

### 4.1. ZMD :

- La balise inlineCode dans rebber [↗](#)
- Amélioration de l'utilisation de yarn et lerna [↗](#)
- Migration du serveur zmd vers le projet zmd [↗](#)

### 4.2. Zds site

- Change le nom d'une erreur pour se préparer à django 1.11 [↗](#)
- Corrige l'affichage des sous-titres des sujets du forum [↗](#)
- Corrige le problème d'encodage dans l'historique des tuto [↗](#)
- Améliore les messages d'erreurs en cas d'échec de login [↗](#)

#### 4. Les PR de la quinzaine

### 4.3. Latex-template

- Ajoute un mode "imprimable" noir et blanc pour le PDF [↗](#) (notons que ceci sera supporté côté serveur via un PDF nommé `{slug-du-tuto}.printable.pdf`)