



Beste de savoir

Action Text : un nouveau composant
pour Ruby on Rails

21 décembre 2018

Table des matières

1. Introduction	1
2. Qu'apporte Action Text ?	1
3. Les critiques à l'encontre d'Action Text	2
4. Conclusion	2

1. Introduction

Le célèbre *framework web* Ruby on Rails ne cesse d'évoluer et de proposer de plus en plus de fonctionnalités au fur et à mesure du temps.

La version 5.0 a par exemple introduit [le composant Action Cable](#) , qui facilite la création de fonctionnalités en temps réel, puis la version 5.1 a permis [l'intégration des outils Yarn et Webpack](#) .

Enfin, la dernière version à ce jour, la 5.2.x, a introduit un nouveau composant, [Active Storage](#) , qui permet la prise en charge du téléchargement de fichiers (images notamment) vers le serveur.

La prochaine version majeure (6.x) du *framework* se profile pour l'horizon 2019, et l'arrivée d'un nouveau composant vient tout juste d'être annoncée. Son nom ? Action Text.

2. Qu'apporte Action Text ?

C'est sur [le blog officiel de Ruby on Rails](#) que ce nouveau venu a été annoncé hier, le 3 octobre 2018.

L'objectif visé est très clair, faciliter l'intégration d'un éditeur de texte **WYSIWYG** (ou éditeur de texte enrichi), qui permet de voir dès la rédaction d'un contenu, son apparence visuelle finale. Parmi ce type d'éditeur, on peut citer parmi les plus connus, Ckeditor, TinyMCE, Froala ou encore [Trix](#) . C'est celui-ci qui nous intéresse, nous allons y revenir.

Évidemment, ce nouveau composant tirera pleinement profit d'Active Storage, ce qui permettra d'envoyer une ou plusieurs images vers le serveur d'un simple glisser-déposer dans le champ texte.

À l'heure actuelle, Action Text est encore en alpha et dans [un dépôt qui lui est propre](#) , séparé du code source de Rails. Néanmoins, le code est déjà utilisé sur [Basecamp](#) depuis plusieurs années, et une vidéo de démonstration est déjà disponible.

<https://youtu.be/HJZ9TnKrt7Q>

3. Les critiques à l'encontre d'Action Text

Bien que l'annonce d'Action Text soit très récente, des critiques sont déjà formulées à son encontre. Il y en a 2 principales.

La première, c'est le manque de polyvalence du nouveau composant. Certains regrettent qu'il soit compatible uniquement avec Trix, et auraient aimé pouvoir utiliser n'importe quel autre éditeur de texte.

Le second est probablement un débat vieux comme les *framework web*. En effet, si une partie de la communauté considère Action Text comme une avancée majeure et un gain de temps, une autre partie considère que cela alourdit Rails, et surtout que ce n'est pas le rôle d'un *framework web* de proposer ce type de fonctionnalités par défaut.

Les premiers arguent que c'est une fonctionnalité très commune sur les sites web aujourd'hui, et que de ce fait, c'est aussi le rôle du *framework* de proposer d'en faciliter l'intégration, le développement et l'usage. En outre, ils avancent qu'il sera probablement possible d'activer ou non cette fonctionnalité en ligne de commandes lors de la création d'un nouveau projet, comme c'est d'ailleurs le cas pour l'utilisation de Webpack par exemple.

Les seconds estiment qu'il s'agit là d'une problématique qui concerne le *front-end*, avec un choix qui ne devrait pas être imposé, aussi bien concernant l'éditeur, que plus généralement concernant le choix technique d'utiliser un éditeur *WYSIWYG* ou non.

4. Conclusion

Même si je pense connaître la pensée majoritaire à ce sujet ici, cette conclusion me permet d'ouvrir le débat.

Que pensez-vous de ce composant à venir ? Le rôle du *framework web* est-il, selon vous, outre-passé ?

C'est la fin de ce très court et premier article de ma part sur ZdS, qui, outre la volonté de partager une information, avait aussi pour but de m'essayer à la rédaction sur ce site, chose que j'hésitais à faire depuis quelque temps déjà.

Par ailleurs, étant donné que je ne suis pas particulièrement habitué à rédiger ce type de contenu à destination du public, je suis également intéressé par tout retour (clarté de la syntaxe, manque ou excès de ponctuation, phrases trop lapidaires ou trop longues, etc.) qui pourrait permettre de m'améliorer.

Liste des abréviations

WYSIWYG What You See Is What You Get. [1](#), [2](#)