

Beste de savoir

[Chronique]Zest Of Dev 2

15 janvier 2019

Table des matières

1.	Introduction	1
2.	Avancement de ZMD et des publications PDF	1
3.	Les tests sont parmi nous	2
3.1.	fin à fin, c'est smoothafin	2
3.2.	nous vous notifions que vous n'êtes plus notifié	2
4.	Les autres news du dev	3
5.	Conclusion	4

% [CHRONIQUE]ZEST OF DEV 2 % artragis % 31 décembre 2017

1. Introduction

2018 approche les amis. Avec un petits jour de retard (merci cher rhume, tu m'as bien ralenti), voici le second épisode de la chronique des devs du site.

Et on a de quoi lever les flutes à smoothie : la v26 a été déployée, c'est donc python3 qui "propulse" le site, pour ne pas utiliser d'anglicisme.

2. Avancement de ZMD et des publications PDF

La v26 est publiée !

du coup le prochain milestone annonce qu'on aura le nouveau moteur de markdown et de génération de PDF.

Actuellement, le développement ne bougeait plus beaucoup car sandhose (qui gère les mises à jour de zmd) était occupé à migrer le serveur et à mettre en place la v26.

Les choses vont donc changer. Quelques difficultés sont néanmoins encore à prévoir :

- le template lualatex intègre une dépendance à un dépôt externe, il faut pouvoir l'installer
- Héliode a fait un bon boulot pour créer une version imprimable du PDF. Mais rien n'est encore codé pour l'utiliser (bien que ça sera rapide).

3. Les tests sont parmi nous

3. Les tests sont parmi nous

En ce moment le dev de nouvelles fonctionnalités bat son plain, alors je pense qu'il peut être sympa de faire une pause sur un point qui permet à zeste de savoir de bugger *relativement peu* au fil des versions : nos tests.

Quand on corrige un bug, on crée un test, cela nous permet de comprendre ce qui bug et aussi de nous assurer qu'on ne casse pas tout à chaque fois¹.

3.1. fin à fin, c'est smoothafin

La v26 a ajouté un élément très important dans notre système de tests : les tests "end to end" qui simule une vraie navigation sur l'interface graphique ! Pour cela on se base sur [selenium](#) [↗](#) .

Et quoi de mieux pour illustrer cela que la [PR de motet-a](#) [↗](#) pour ajouter la persistance des textes dans nos zones de saisies ?

Motet-a a donc écrit une fonctionnalité bien sympathique : vous commencez à écrire du texte sur votre zones actuelle, par exemple, pour un tuto. Là, pour une raison ou une autre vous devez naviguer sur le site (ajout d'une image dans la galerie, retour dans l'historique...) et vous quittez la page actuelle : vous venez de perdre votre texte. Enfin, ça c'était avant la modification de motet-a.

Mais comment s'assurer que dans le futur, on ne cassera pas cette fonctionnalité ? Vous l'avez deviné, en écrivant un test.

Pour l'instant le test de motet-a souffre du syndrome "chez moi, ça marche", mais on est dessus, ne vous inquiétez pas.

3.2. nous vous notifions que vous n'êtes plus notifié

L'arrivée des tests avec interface graphique ne signifie pas l'abandon des anciens tests et actuellement ce sont sûrement les tests à propos des notifications qui sont le meilleur exemple.

En effet les notifications souffrent souvent de légers bugs de "notification persistante mais facile à désactiver" du fait que lorsqu'un élément est supprimé/masqué/déplacé, il devient inaccessible alors que la notification qui lui est associé n'est pas... notifiée de ce changement.

Ici il s'agira principalement de décrire en python le scénario que les utilisateurs nous rapporte. Par exemple :

Lorsqu'un auteur publie un billet, que je m'inscrit à ce billet, qu'un commentaire est posté, si l'auteur dépublie le billet avant que j'ai eu le temps de lire le commentaire, j'ai une notification persistante.

devient :

1. comme je l'expliquais dans [mon dernier article](#) [↗](#) .

4. Les autres news du dev

```
1 def test_no_persistent_comment_notif_on_revoke(self):
2     from zds.tutorialv2.publication_utils import
3         unpublish_content
4     content = PublishedContentFactory(author_list=[self.user2],
5         type='OPINION') # créons un billet publié par user2
6
7     ContentReactionAnswerSubscription.objects.get_or_create_active(self
8     content) # user1 souscrit au billet
9     ContentReactionFactory(related_content=content,
10        author=self.user2, position=1) # user2 poste un message
11     self.assertEqual(1,
12        len(Notification.objects.get_unread_notifications_of(self.user1)))
13        # on s'assure que user1 a été notifié
14     unpublish_content(content, moderator=self.user2) # on
15        dépublie le billet
16     self.assertEqual(0,
17        len(Notification.objects.get_unread_notifications_of(self.user1)))
18        # on s'assure qu'il n'y a plus de notification non lue,
19        conformément à ce qu'attend l'utilisateur
```

avant de fixer le bug, le test échoue. Après, il réussit. Tada, nous avons corrigé un problème de notification et nous sommes assuré que dans le futur, rien en serait cassé.

4. Les autres news du dev

Cette semaine fut l'occasion d'accueillir parmi les contributeurs du projet un petit nouveau : nils-van-zuijlen. Bienvenue à lui!

Sa [première PR](#) a été fusionnée au code commun et il [continue sur sa lancée](#).

Eskimon et Anto ont quant à eux repris le travail sur les [statistiques](#) et il n'y a pas à dire, les premiers brouillons envoient du lourd.

Plusieurs PR sont proches d'être *mergées*, il ne manque plus qu'un peu de tests pour s'assurer que tout va bien :

- [Une API pour les contenus](#)
- [Correction d'un lien dans la page de recherche](#)
- [Faciliter la mise en une d'un contenu pour le staff](#)

Côté "fonctionnalités à venir", le concept pour [la publication partielle](#) semble avoir été approuvé, je reprendrai le travail rapidement.

Héziode a d'ailleurs terminé de corriger les problèmes relatifs à la page de couverture.

5. Conclusion

5. Conclusion

Deux belles semaines à coder, avec un nouveau contributeur, voilà du bon pour démarrer cette année 2018.

Meilleurs voeux à tous !