ARTRAGIS

Zeste de savoir

[chronique] Zest of dev 12

21 janvier 2019

Table des matières

1. Introduction

Bonjour à tous et bienvenu dans ce nouvel épisode de Zest of Dev : la chronique du développement de Zeste de Savoir. Une chronique qui innove car pour la première fois je la tappe en bépo. C'est peut-être un détail pour vous mais pour moi ça veut dire beaucoup.

Au programme une réunion manquée, une migration en douceur et des PDF en veux-tu en voilà !

2. Zest meeting où es-tu ?

Dans le dernier zest of dev je vous avais parlé d'un «zest meeting» à venir.

J'ai tenté de l'organiser et disons que ça a fait un four.

Je suis donc désolé mais de ce point de vue je n'aurai pas de nouvelle à vous donner.

FIGURE 2. – désolé

3. Django django eh eh

Zeste de savoir est développé grâce au framework `django`.

Ce framwork est totalement open-source et il est activement développé. Cela entraîne certaines conséquences assez positives :

— Il y a fréquemment des nouvelles fonctionnalités qui sont ajoutées ;
— Une veille de qualité est opérée pour assurer la sécurité du framework.

Et de notre côté nous essayons de suivre les mises à jour.

La version 28 de zds utilise la branche 1.11 pour fonctionner mais d'ici quelques temps cette branche ne sera plus supportée. Alors j'ai entrepris, pour la version 28.1 de passer à la version 2.1 de django : voici mon histoire.

Première remarque la documentation de django est assez complète et on le verra plus tard je n'ai eu à galérer que sur un seul point somme toute assez minime. Je remercie donc l'équipe de développement de django, ils font un travail formidable.

3.1. Level un : pester contre la procrastination

Django est sympa, il vous prévient super en avance quand une fonctionnalité est vouée à disparaître ou qu'elle sera améliorée mais sans pouvoir assurer la rétrocompatibilité. Vous avez des jolis avertissement et vous vous dîtes « Plus tard » et vous finissez par les faire taire. Et quand la migration arrive et que certaines fonctions qui avaient des paramètres optionnels vous oblige désormais à les passer, vous n'avez plus que vos yeux pour pleurer.

Ainsi il m'aura fallu deux heures pour mettre d'équerre notre modèle de données, une autre heure pour que les URLs soient fonctionnelles (mais avec un gros avertissement car on avait vraiment mal fait les choses.

D'ailleurs ça a été pour moi l'occasion de découvrir l'excellent module `django-extensions` .

Le premier niveau est passé, mais l'application ne démarre toujours pas.

3.2. Level 2 : Dépendances et intergiciels

J'adore ce mot, pas vous ?

django-social-auth (la dépendance legacy pour l'autentification par les réseaux) n'est pas et ne sera jamais compatible django 2.X.

Heureusement que les développeurs de son remplaçant ont écrit un manuel de migration.

Et vous savez quoi ZDS a trois composanti `middlewares` (intergiciels). Mais y'en a un il est caché dans un fichier init. Et comme django 2.0 a changé la manière d'instantier les middlewares django était pas content de la manière dont on avait fait les choses MAIS PAS MOYEN DE TROUVER ce middleware manquant !

Finalement tout s'est bien passé, la mère et l'enfant vont bien... Quatre tests sont au rouge sur les plus de 600 du site.

3.3. PEBKBC et autres convives

Je vous l'ai dit il a fallu ajouter des argument obligatoires à nos modèles. Plus précisemment il fallait préciser à django que faire lorsqu'on supprime un objet qui est référencé par un autre dans le cas d'une `foreignKey` ou une `OneToOne`.

Avant si vous ne disiez rien django comprenait par défaut « suppression en cascade ». Là il fallait le dire explicitement. Je suis donc repassé sur tous nos modèles pour mettre `on_delete=models.CASCADE` où il le fallait.

Et puis en passant j'ai changé le comportement où c'était pertinent (me disais-je). Et une fois j'ai été trop gourmand et cela a causé l'échec d'un test. Vite corrigé. Mais ne faites pas comme moi si un jour vous êtes dans une situation similaire.

Le test suivant est dû au seul oubli que j'ai détecté dans la doc de django et encore je ne leur en tient pas trop rigueur.

Pour aficher vos pages HTML django intègre un moteur de template. Ce moteur est extensible notamment grâce aux templatetags. Pour l'un d'entre eux , avions suivi un conseil glâné sur internet qui nous a induit à utiliser la constante `django.base.text.TOKEN_TEXT`.

Seulement un coup d'oeil à la documentation du module dit, en gros « Dans ce module vous n'avez besoin que de `Template`» Et comme entre django 2.0 et 2.1 la constante a été remplacée par une enum bah chez nous ça pétait.

Pour les deux autres tests je n'approfondis pas car ce billet est déjà assez long et qu'ijs sont dû à des erreurs de logique qui n'ont pas grand chose à voir avec la migration.

3.4. Et maintenant ?

Bah faut tester.

4. Les PDFs

Nos exports en PDFs sont désormais de bonne qualité et je fais de mon mieux pour rattraper le retard dans leur génération.

En attendant un topic spécifique a été ouvert https ://zestedesavoir.com/forums/sujet/11965/export-pdf-revolution-et-bugs-connus