

Chrono - Development #30117

Encapsuler les vue dans une transaction

25 janvier 2019 19:00 - Emmanuel Cazenave

Statut:	Rejeté	Début:	25 janvier 2019
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Dans #30085 un Event sans Booking associé.			
Dans l'API, fillslot créé l'évènement, plus loin le booking, donc il y de la place pour que ça merdouille. Pas trouvé la trace donc ce n'est peut-être pas ça mais c'est de toute façon une bonne pratique.			

Historique

#1 - 25 janvier 2019 19:20 - Frédéric Péters

Il y a moyen de limiter ça à l'endroit concerné ? (pour ne pas ajouter un comportement par défaut qui sera différent des autres modules Publik).

(et quand même, en l'état, côté debian/, ça va virer le contenu utile de DATABASES et plus rien ne va marcher.)

#2 - 25 janvier 2019 19:43 - Emmanuel Cazenave

- Fichier *0001-wrap-every-view-in-a-transaction-30117.patch* ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Tu me devances de quelques minutes, j'allais proposer ça.

Je ne vais pas en faire en cheval de bataille, mais ça me semblerait intéressant de tester l'encapsulation globale sur une brique plutôt que d'encapsuler par ci par là au fil de l'eau, je ne vois pas dans quels cas une écriture partielle serait bienvenue et si ça foire il est facile de revenir en arrière.

(j'ai été surpris il y a quelques mois en découvrant que ce n'était pas le comportement par défaut de django, j'étais habitué à des framework qui font le choix inverse)

#3 - 25 janvier 2019 20:56 - Frédéric Péters

sur une brique

Chrono n'est pas volontaire.

#4 - 25 janvier 2019 20:59 - Frédéric Péters

Pour expérimenter, on peut voir pour poser un fichier dans les settings.d/, avec `DATABASES['default']['ATOMIC_REQUESTS'] = True` dedans, pour toutes les briques.

#5 - 25 janvier 2019 21:28 - Benjamin Dauvergne

À vrai dire c'était le cas sur tous nos logiciels avant Django 1.6 avec le le middleware `TransactionMiddleware`; i.e. avant la migration portail-citoyen2 vers combo. Ça aurait du rester le défaut sur a2 (j'ai `ATOMIC_REQUESTS` dans le settings de base) mais j'ai raté ces 3 dernières années le fait que hobo a surchargé ça, et puis je ne m'en suis pas soucier plus que ça). Je suis volontaire coté A2 pour revenir à la situation antérieure, ça peut potentiellement créer des contentions qu'on ne voyait pas avant, voir faire des erreurs au commit, mais je m'inquiéterais plus si on faisait ça coté w.c.s. qui fait beaucoup plus de manipulation en base que nos autres logiciels; sur combo ce serait assez indolore parce qu'il est read-only et sur passerelle on a aussi très peu d'action concurrentes donc idem peu de soucis envisageables.

#6 - 25 janvier 2019 21:39 - Benjamin Dauvergne

Coté chrono il n'y a qu'un seul endpoint qui fait des écritures, fillslot, on pourrait se contenter d'un `@atomic()`, coté model il semble y a voir déjà pas mal d'`atomic()` utilisés mais on pourrait faire le tour.

#7 - 26 janvier 2019 09:53 - Frédéric Péters

- Statut changé de *Solution proposée* à *Rejeté*

Dans l'API, fillslot crée l'évènement, plus loin le booking, donc il y a de la place pour que ça merdouille.

J'ai lu le code, et non : l'Event est créé avec full=False, et le Booking + l'update pour faire full=True sont déjà dans une transaction (dans le .save() de Booking).

Je laisse du coup la discussion se porter à un niveau plus général. (j'ai ajouté quelques lignes en préparation de la réunion de lundi).

Quant au bug de #30085, il doit être traité en tant que tel, voir comment un objet Booking a pu être supprimé, alors qu'aucun code dans Chrono ne fait (explicitement) ça.

Fichiers

0001-wrap-every-view-in-a-transaction-30117.patch

989 octets

25 janvier 2019

Emmanuel Cazenave