

Authentic 2 - Development #47391

Désactiver la recherche approximative si immutable_unaccent est indisponible

06 octobre 2020 15:07 - Benjamin Dauvergne

Statut:	Rejeté	Début:	06 octobre 2020
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Pour éviter les catastrophes.			

Historique

#1 - 06 octobre 2020 15:07 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#2 - 06 octobre 2020 15:43 - Benjamin Dauvergne

- Fichier 0003-misc-birhtdate-attribute-content-must-be-serialized-.patch ajouté
- Fichier 0001-misc-add-tool-to-check-existence-of-SQL-functions-47.patch ajouté
- Fichier 0002-misc-use-a-fallback-if-trigram-search-is-not-possibl.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

#3 - 06 octobre 2020 16:51 - Valentin Deniaud

Pourquoi 0003 ? Ça semble marcher sans, il y a un test.

#4 - 06 octobre 2020 19:22 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Pourquoi 0003 ? Ça semble marcher sans, il y a un test.

Qu'est-ce qui marche sans quoi ? Sans pg_trgm et immutable_unaccent peut-on utiliser find_duplicates() partout ?

#5 - 07 octobre 2020 10:09 - Valentin Deniaud

Benjamin Dauvergne a écrit :

Valentin Deniaud a écrit :

Pourquoi 0003 ? Ça semble marcher sans, il y a un test.

Qu'est-ce qui marche sans quoi ? Sans pg_trgm et immutable_unaccent peut-on utiliser find_duplicates() partout ?

Je parle du patch numéro 3 pas du ticket en général, misc: birhtdate attribute content must be serialized, pourquoi il y a eu besoin de modifier cette ligne ?

#6 - 08 octobre 2020 13:41 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Benjamin Dauvergne a écrit :

Valentin Deniaud a écrit :

Pourquoi 0003 ? Ça semble marcher sans, il y a un test.

Qu'est-ce qui marche sans quoi ? Sans pg_trgm et immutable_unaccent peut-on utiliser find_duplicates() partout ?

Je parle du patch numéro 3 pas du ticket en général, misc: birthdate attribute content must be serialized, pourquoi il y a eu besoin de modifier cette ligne ?

C'est parce que ça dépend d'un effet de bord pour fonctionner dont je me méfie, la colonne content c'est un CharField mettre un datetime.date dedans ça marche parce qu'implicitement, je ne sais trop qui, de l'ORM ou de psycopg2, caste la valeur en str et que par défaut str(date) ça donne bien le format %Y-%m-%d qu'on attend. Mais je préfère qu'on soit explicite ici pour le jour on en changerait la façon de stocker ça et qu'on voudrait vraiment un date alors que des gens se sont mis à passer directement une chaîne parce que ça marche.

#7 - 08 octobre 2020 15:07 - Valentin Deniaud

Oui OK du coup j'ai fouillé pour savoir comment ça marchait quand on faisait user.birthdate = datetime(...), c'est dans attribute_kinds.py

```
267         'serialize': lambda x: x and x.isoformat(),
```

Et donc plutôt que strftime tape un isoformat et je serai content :)

Mais à relire les patches d'avant, l'approche choisie me fait assez peur :

- Imagine si la première version du patch incluait ça, on aurait mis combien de temps à se rendre compte que les extensions n'avaient pas été installées ? Si crasher c'est pas bien, il faut quand même un mail.
- Et surtout, avoir du sql brut c'est relou à maintenir surtout vu la tête de la requête. Et le jour où une màj fait que la requête ne marche plus on va mettre des mois avant de s'en rendre compte.

À la place je verrai bien un try/except, qui en cas de ProgrammingError fasse un logger.exception et retourne find_duplicate_slow.

#8 - 08 octobre 2020 15:52 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Oui OK du coup j'ai fouillé pour savoir comment ça marchait quand on faisait user.birthdate = datetime(...), c'est dans attribute_kinds.py [...]

Et donc plutôt que strftime tape un isoformat et je serai content :)

Ok je suis allé chercher directement la définition ça évite de se demander d'où ça vient.

Mais à relire les patches d'avant, l'approche choisie me fait assez peur :

- Imagine si la première version du patch incluait ça, on aurait mis combien de temps à se rendre compte que les extensions n'avaient pas été installées ? Si crasher c'est pas bien, il faut quand même un mail.

Ce n'est pas moi qui ait choisi cette approche du "le jeudi rien ne doit arriver"; je fais avec.

- Et surtout, avoir du sql brut c'est relou à maintenir surtout vu la tête de la requête. Et le jour où une màj fait que la requête ne marche plus on va mettre des mois avant de s'en rendre compte.

Je ne sais pas de quel SQL tu parles exactement mais le fait que l'ORM rende le SQL abstrait c'est une vue de l'esprit, on est bloqué sur postgres maintenant ; mais bon c'est comme ça qu'on fait en Django, les gens qui font du hibernate ou symphony ne font pas comme ça et ça se passe très bien quand même.

À la place je verrai bien un try/except, qui en cas de ProgrammingError fasse un logger.exception et retourne find_duplicate_slow.

Je ne vois pas ce que ça apporte. L'erreur en cas d'absence de l'extension est certaine; là tu parles d'une erreur qu'ont aurait pas encore identifié, on peut mettre des try/except autour de toutes les lignes de code et en proposer deux implémentations mais ça va nous prendre du temps. Les traces qui font une 500 on les corrige plus vite que les traces qui n'en font pas :)

One fois la première mise en production faite on devrait retirer le code qui empêche la migration d'installation de l'extension de crasher, on ne la fera tourner que sur des nouveaux déploiements qui généralement ne se font pas dans l'urgence. Dans ce cas on pourrait **fermer** ce ticket qui devient inutile. J'essaie juste de contourner l'état incertain dans lequel nous met le fait de cacher les erreurs d'installation des extensions.

#9 - 08 octobre 2020 16:08 - Frédéric Péters

Ce n'est pas moi qui ait choisi cette approche du "le jeudi rien ne doit arriver"; je fais avec.

"Le jeudi les mises à jour doivent pouvoir se faire sans problèmes" c'est quelque chose derrière quoi tout le monde devrait se rallier, pas "faire avec"; déjà qu'elles doivent se faire à une heure de plus en plus tardive, s'il fallait en plus gérer des problèmes de mises à jour et/ou devoir différencier les situations façon "là c'est bon la manip sur le postgresql a été réalisée", ça deviendrait vraiment difficile.

#10 - 08 octobre 2020 16:12 - Benjamin Dauvergne

Frédéric Péters a écrit :

"Le jeudi les mises à jour doivent pouvoir se faire sans problèmes" c'est quelque chose derrière quoi tout le monde devrait se rallier, pas "faire avec"; déjà qu'elles doivent se faire à une heure de plus en plus tardive, s'il fallait en plus gérer des problèmes de mises à jour et/ou devoir différencier les situations façon "là c'est bon la manip sur le postgresql a été réalisée", ça deviendrait vraiment difficile.

Je ne dis pas qu'il faut se foutre des gens qui font les mises à jour; je dis qu'il y a plusieurs façon de faire, on aurait pu faire un migrate_schemas --fake custom_user xxxx pour les quelques bases où on aura pas obtenu les droits dans les temps après l'erreur plutôt que de cacher le problème sous le tapis ad-vitam. Il faut arrêter de surinterpréter ce que je dis.

#11 - 08 octobre 2020 16:17 - Valentin Deniaud

Benjamin Dauvergne a écrit :

- Imagine si la première version du patch incluait ça, on aurait mis combien de temps à se rendre compte que les extensions n'avaient pas été installées ? Si crasher c'est pas bien, il faut quand même un mail.

Ce n'est pas moi qui ait choisi cette approche du "le jeudi rien ne doit arriver"; je fais avec.

Wut ? Pour reformuler, sur un déploiement où les extensions ne se sont pas installées, on en sera averti pas une 500 sur /find_duplicates/. Avec ce patch on fallback silencieusement sur find_duplicate_slow, et on ne le saura « jamais » : il me semble que c'est un problème.

- Et surtout, avoir du sql brut c'est relou à maintenir surtout vu la tête de la requête. Et le jour où une maj fait que la requête ne marche plus on va mettre des mois avant de s'en rendre compte.

Je ne sais pas de quel SQL tu parles exactement

Celui que tu ajoutes dans 0001.

À la place je verrai bien bien un try/except, qui en cas de ProgrammingError fasse un logger.exception et retourne find_duplicate_slow.

Je ne vois pas ce que ça apporte.

De se passer du sql.

J'essaie juste de contourner l'état incertain dans lequel nous met le fait de cacher les erreurs d'installation des extensions.

Peut-être que je n'ai simplement pas saisi le réel intérêt de ce ticket, tu parles de « catastrophe » dans la description et je ne vois pas de quoi il s'agit.

#12 - 08 octobre 2020 16:39 - Benjamin Dauvergne

- Statut changé de Solution proposée à Rejeté

Bon ok ça crashera.

Fichiers

0003-misc-birhtdate-attribute-content-must-be-serialized-.patch	1,05 ko	06 octobre 2020	Benjamin Dauvergne
0001-misc-add-tool-to-check-existence-of-SQL-functions-47.patch	3,11 ko	06 octobre 2020	Benjamin Dauvergne
0002-misc-use-a-fallback-if-trigram-search-is-not-possibl.patch	4,41 ko	06 octobre 2020	Benjamin Dauvergne