

Passerelle - Development #42312

Pouvoir utiliser le JSONField de postgres

30 avril 2020 10:22 - Valentin Deniaud

Statut:	Fermé	Début:	30 avril 2020
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Dans [#42008](#), on se rend compte que ce n'est pas possible car django-jsonfield < 1.3 est buggé et empêche d'utiliser les deux en parallèle.

Les tests passent avec la dernière version de django-jsonfield, mais les paquets debian sont plutôt autour de la version 1.1.

Alternativement, virer la dépendance avec django-jsonfield et tout faire avec postgres.

Ça a été fait dans a2, [#29193](#), mais il y avait volonté de supporter sqlite, donc sans cette contrainte peut-être qu'il y a plus simple que ce qui est dit dans le ticket.

Révisions associées

Révision 5e1c910f - 04 mai 2020 18:30 - Valentin Deniaud

misc: use postgres JSONField (#42312)

Révision 0c5cad22 - 04 mai 2020 18:30 - Valentin Deniaud

misc: add missed migrations (#42312)

Révision 9ef99310 - 05 mai 2020 10:07 - Valentin Deniaud

misc: add migration to ensure jsonb type (#42312)

Historique

#1 - 30 avril 2020 10:26 - Valentin Deniaud

À la suite du début des discussions dans [#42008](#),

Frédéric Péters a écrit :

Il me semble qu'on ne fait vraiment rien de particulier, qu'il n'y aurait même pas besoin de migration (juste traffiquer une précédente pour faire croire qu'on a depuis toujours utiliser le jsonfield de django.contrib.postgres).

Il y a effectivement pas grand chose à faire, mais au moins un ALTER TABLE, pour changer le type de la colonne de json à jsonb.

Une manière de faire déjà un peu lourde est présentée ici <https://tech.polyconseil.fr/migrating-from-json-to-jsonb.html>.

#2 - 30 avril 2020 10:29 - Frédéric Péters

Mais, c'est déjà jsonb : (j'ai regardé la prod, et chez moi) (?)

Colonne	Type	Modificateurs
id	integer	non NULL Par défaut, nextval('opengis_featurecache_id_seq'::regclass)
lat	double precision	non NULL
lon	double precision	non NULL
data	jsonb	non NULL
query_id	integer	non NULL
text	character varying(2048)	non NULL

#3 - 30 avril 2020 11:01 - Valentin Deniaud

OK, je comprends pas le pourquoi du comment, peut-être que Benj en saura plus.

#4 - 30 avril 2020 11:09 - Lauréline Guérin

dans django-jsonfield:

```
def db_type(self, connection):
    if connection.vendor == 'postgresql':
        # Only do jsonb if in pg 9.4+
        if connection.pg_version >= 90400:
            return 'jsonb'
        return 'text'
```

Heureusement qu'on n'a pas du text en prod :)

#5 - 30 avril 2020 12:43 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

#6 - 30 avril 2020 12:43 - Valentin Deniaud

- Fichier 0001-misc-use-postgres-JSONField-41992.patch ajouté

- Tracker changé de Support à Development

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Bien vu ! Donc la seule chose qui va casser, c'est quand on essaiera de créer un objet sans remplir le JSONField, parce que django-jsonfield posait dynamiquement un default=dict. Il y a deux endroits qui ont été identifiés dans les tests et que j'ai corrigés, mais sûrement qu'il y en a d'autre.

Et bien sûr, si il existe une prod datant d'avant postgres 9.4 ça va exploser en mille morceaux, mais peut-être que c'est imaginable de corriger à la main/faire tourner un script dans ces cas là.

#7 - 30 avril 2020 12:49 - Frédéric Péters

Et bien sûr, si il existe une prod datant d'avant postgres 9.4 ça va exploser en mille morceaux, mais peut-être que c'est imaginable de corriger à la main/faire tourner un script dans ces cas là.

On peut imaginer une migration qui assure que c'est déjà jsonb, qui fasse le changement sinon ? Parce que ça va arriver et c'est quand même bien plus facilement gérable partout si c'est géré via une migration.

Table « passerelle_vincennes_entrouvert_com.csvdatasource_csvdatasource »		
Colonne	Type	Modificateurs
id	integer	non NULL Par défaut, ...
title	character varying(50)	non NULL
slug	character varying(50)	non NULL
description	text	non NULL
csv_file	character varying(100)	non NULL
columns_keynames	character varying(256)	non NULL
skip_header	boolean	non NULL
_dialect_options	text	<-----
sheet_name	character varying(150)	non NULL

#8 - 30 avril 2020 15:10 - Valentin Deniaud

Frédéric Péters a écrit :

On peut imaginer une migration qui assure que c'est déjà jsonb, qui fasse le changement sinon ? Parce que ça va arriver et c'est quand même bien plus facilement gérable partout si c'est géré via une migration.

Oui donc on retombe tout à fait sur la méthode du billet de blog cité plus haut, sauf qu'on s'autorise de changer le code et la db en même temps.

Tu écris une migration donc pour être sûr que je ne me lance pas sur une fausse piste, on parle d'écrire à la mano une dizaine de fichiers de migrations, soit un par app qui utilise un jsonfield, à base de RunSQL ?

#9 - 30 avril 2020 16:09 - Frédéric Péters

Tu écris une migration donc pour être sûr que je ne me lance pas sur une fausse piste, on parle d'écrire à la mano une dizaine de fichiers de migrations, soit un par app qui utilise un jsonfield, à base de RunSQL ?

Oui; en modérant le côté "une dizaine de fichiers", genre en créant notre propre classe de EnsureJsonbType(Operation), pour tout à fait réduire la duplication, genre ces migrations seraient juste :

```
operations = [  
    EnsureJsonbType(model_name='csvdatasource', field_name='_dialect_options'),  
]
```

#10 - 04 mai 2020 17:26 - Valentin Deniaud

- Fichier 0002-misc-add-missed-migrations-42312.patch ajouté
- Fichier 0003-misc-add-migration-to-ensure-jsonb-type-42312.patch ajouté
- Fichier 0001-misc-use-postgres-JSONField-42312.patch ajouté

Voilà pour la migration partout. Il faut que j'ajoute un test ?

#11 - 04 mai 2020 17:52 - Frédéric Péters

Si ça te semble possible, oui un test pour la migration de passerelle/utills/db.py ce serait super, mais si c'est galère parce que ça oblige à sortir de l'ORM pour traffiquer la DB, on peut juste être très attentif en relisant...

(mais il faut ajouter l'entête de licence au fichier)

#12 - 04 mai 2020 18:31 - Valentin Deniaud

- Fichier 0002-misc-add-missed-migrations-42312.patch ajouté
- Fichier 0003-misc-add-migration-to-ensure-jsonb-type-42312.patch ajouté
- Fichier 0001-misc-use-postgres-JSONField-42312.patch ajouté

Voilà un test, c'était pas si horrible, juste un petit peu.

Pas besoin de tester toutes les migrations je pense, le migrate exécuté de toute façon par les tests garanti normalement l'absence de faute de frappe sur le nom des tables/colonnes.

#13 - 04 mai 2020 19:14 - Frédéric Péters

```
reversible = False
```

Je mettrais True, et un backwards qui ne fait rien vu que ça vivait déjà très bien en jsonb; non ?

#14 - 05 mai 2020 10:08 - Valentin Deniaud

- Fichier 0002-misc-add-missed-migrations-42312.patch ajouté
- Fichier 0003-misc-add-migration-to-ensure-jsonb-type-42312.patch ajouté
- Fichier 0001-misc-use-postgres-JSONField-42312.patch ajouté

Oui tout à fait.

#15 - 05 mai 2020 10:18 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

#16 - 05 mai 2020 11:18 - Valentin Deniaud

- Statut changé de Solution validée à Résolu (à déployer)

```
commit 9ef993108b945a41d2f9d430c6820c6c90cae6b0  
Author: Valentin Deniaud <vdeniaud@entrouvert.com>  
Date: Mon May 4 14:04:32 2020 +0200
```

```
misc: add migration to ensure jsonb type (#42312)
```

```
commit 0c5cad2296914a9d2a4865accd391eed507fdb55  
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
```

Date: Mon May 4 14:03:41 2020 +0200

misc: add missed migrations (#42312)

commit 5e1c910fbfacb15dd37ea8f0192561dcc15054bc
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Apr 30 12:00:34 2020 +0200

misc: use postgres JSONField (#42312)

#17 - 07 mai 2020 16:16 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Solution déployée

Fichiers

0001-misc-use-postgres-JSONField-41992.patch	43,9 ko	30 avril 2020	Valentin Deniaud
0002-misc-add-missed-migrations-42312.patch	15,5 ko	04 mai 2020	Valentin Deniaud
0003-misc-add-migration-to-ensure-jsonb-type-42312.patch	15,8 ko	04 mai 2020	Valentin Deniaud
0001-misc-use-postgres-JSONField-42312.patch	43,9 ko	04 mai 2020	Valentin Deniaud
0002-misc-add-missed-migrations-42312.patch	15,5 ko	04 mai 2020	Valentin Deniaud
0003-misc-add-migration-to-ensure-jsonb-type-42312.patch	18,8 ko	04 mai 2020	Valentin Deniaud
0001-misc-use-postgres-JSONField-42312.patch	43,9 ko	04 mai 2020	Valentin Deniaud
0002-misc-add-missed-migrations-42312.patch	15,5 ko	05 mai 2020	Valentin Deniaud
0003-misc-add-migration-to-ensure-jsonb-type-42312.patch	18,9 ko	05 mai 2020	Valentin Deniaud
0001-misc-use-postgres-JSONField-42312.patch	43,9 ko	05 mai 2020	Valentin Deniaud