

Authentic 2 - Development #71619

Dépréciation: django 3.1 a déprécié NullBooleanField et mis JSONField dans les champs de modèle génériques

23 novembre 2022 15:19 - Benjamin Dauvergne

Statut:	Fermé	Début:	23 novembre 2022
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description https://docs.djangoproject.com/en/dev/releases/3.1/#id2 https://docs.djangoproject.com/en/dev/releases/3.1/#jsonfield-for-all-supported-database-backends			

Révisions associées

Révision 5788e9f0 - 01 décembre 2022 11:54 - Benjamin Dauvergne

NullBooleanField is deprecated since Django 3.1 (#71619)

It must be replaced by BooleanField(null=True).

Révision 3fcbe219 - 01 décembre 2022 11:54 - Benjamin Dauvergne

django.contrib.postgres's JSONField is deprecated (#71619)

Historique

#1 - 23 novembre 2022 15:57 - Valentin Deniaud

Note, c'est géré par django-upgrade (<https://github.com/adamchainz/django-upgrade#django-31>), tu peux utiliser ça pour faire les adaptations automatiquement si tu veux.

#2 - 23 novembre 2022 16:24 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Note, c'est géré par django-upgrade (<https://github.com/adamchainz/django-upgrade#django-31>), tu peux utiliser ça pour faire les adaptations automatiquement si tu veux.

Par bonne volonté j'ai essayé dans une branche sur le coté mais :

- ça ne corrige pas les migrations
- ça n'ajoute pas de code rétro-compatibilité pour 2.2
- dans le cas de UniqueBooleanField pour lequel je n'ai pas voulu chercher s'il était encore nécessaire ça ne fait rien
- ça n'a pas ajouter un blank=True qui était nécessaire à user_can_reset_password

Je ne pense pas qu'on puisse vraiment utiliser django-upgrade tant qu'on aura besoin de supporter plusieurs versions en parallèle (et pour les migrations je ne comprends pas vraiment pourquoi il ne corrige pas, là l'outil est juste inutile), c'est souvent casse-gueule.

#3 - 23 novembre 2022 16:24 - Benjamin Dauvergne

- Fichier 0001-NullBooleanField-is-deprecated-since-Django-3.1-7161.patch ajouté

- Fichier 0002-django.contrib.postgres-s-JSONField-is-deprecated-71.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#4 - 23 novembre 2022 16:42 - Valentin Deniaud

Benjamin Dauvergne a écrit :

- ça ne corrige pas les migrations

It's a feature je pense, « Transforms the `NullBooleanField()` model field to `BooleanField(null=True)`. Ignores usage in migration files, since Django kept the old class around to support old migrations. »

- ça n'ajoute pas de code rétro-compatibilité pour 2.2

Oui mais ce code n'est nécessaire que parce que tu prends de l'avance, ce n'est pas le principe de `django-upgrade` qui reformate seulement une fois que le support de 2.2 n'est plus nécessaire :)

- ça n'a pas ajouter un `blank=True` qui était nécessaire à `user_can_reset_password`

Ouep c'est le genre de chose à voir à la main, effectivement.

Et remarque à propos du patch, il faut mettre des explicites `if django.VERSION < (3, 1)`: plutôt que gérer les `ImportError`, ça permettra à... `django-upgrade` de les virer automatiquement.

#5 - 23 novembre 2022 17:15 - Benjamin Dauvergne

Valentin Deniaud a écrit :

- ça n'a pas ajouter un `blank=True` qui était nécessaire à `user_can_reset_password`

Ouep c'est le genre de chose à voir à la main, effectivement.

Et remarque à propos du patch, il faut mettre des explicites `if django.VERSION < (3, 1)`: plutôt que gérer les `ImportError`, ça permettra à... `django-upgrade` de les virer automatiquement.

Je prends de l'avance parce que recevoir 20 lignes de warnings lors d'un `make-migrations` c'est fatiguant :) C'est pareil pour les autres patches concernant les warnings, on ne voit plus les erreurs.

Et remarque à propos du patch, il faut mettre des explicites `if django.VERSION < (3, 1)`: plutôt que gérer les `ImportError`, ça permettra à... `django-upgrade` de les virer automatiquement.

Ok je vais faire ça.

#6 - 24 novembre 2022 14:47 - Benjamin Dauvergne

- Fichier `0003-tox.ini-pass-posargs-to-pylint-71619.patch` ajouté
- Fichier `0001-NullBooleanField-is-deprecated-since-Django-3.1-7161.patch` ajouté
- Fichier `0002-django.contrib.postgres-s-JSONField-is-deprecated-71.patch` ajouté

Je n'ai pas trouvé où placé des imports conditionnés sans `pylint` ne chouine d'une manière ou d'une autre, ça s'est fini en `pylint`: `disable=ungrouped-imports` sur une ligne à chaque fois, le moins couteux.

#7 - 01 décembre 2022 10:43 - Valentin Deniaud

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

OK mais de préférence sans 0003, je ne sais pas ce que ça fait là.

#8 - 01 décembre 2022 11:54 - Benjamin Dauvergne

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

```
commit 3fcbe2198c8ef41fc4c74d3397c3d24e272ba6f2
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Wed Nov 23 15:27:27 2022 +0100
```

```
django.contrib.postgres's JSONField is deprecated (#71619)
```

```
commit 5788e9f0cd71047fbd8bb7772dc6b6eb92f11c1e
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Wed Nov 23 15:22:07 2022 +0100
```

```
NullBooleanField is deprecated since Django 3.1 (#71619)
```

```
It must be replaced by BooleanField(null=True).
```

#9 - 23 décembre 2022 10:14 - Transition automatique

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

#10 - 26 février 2023 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-NullBooleanField-is-deprecated-since-Django-3.1-7161.patch	3,57 ko	23 novembre 2022	Benjamin Dauvergne
0002-django.contrib.postgres-s-JSONField-is-deprecated-71.patch	13,2 ko	23 novembre 2022	Benjamin Dauvergne
0003-tox.ini-pass-posargs-to-pylint-71619.patch	720 octets	24 novembre 2022	Benjamin Dauvergne
0001-NullBooleanField-is-deprecated-since-Django-3.1-7161.patch	3,57 ko	24 novembre 2022	Benjamin Dauvergne
0002-django.contrib.postgres-s-JSONField-is-deprecated-71.patch	14,3 ko	24 novembre 2022	Benjamin Dauvergne