

Authentic 2 - Development #57499

Ajouter des champs created et deleted (soft delete) sur Role.members.through

01 octobre 2021 12:52 - Benjamin Dauvergne

Statut:	Nouveau	Début:	01 octobre 2021
Priorité:	Normal	Echéance:	
Assigné à:	Paul Marillonnet	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		
Description On peut visiblement introduire un modèle "trough" sans trop se prendre la tête ¹ . Il faut introduire un champ created en mode auto_now_add=True et un champ deleted nullable. Les retraits d'usager d'un rôle doivent être transformé en soft delete, et les ajouts doivent remettre deleted à None et created à now() si le modèle existe déjà. On doit pouvoir ajouter une property "active_members" qui ne renverrait que les usagers réellement member, tous les accès devront passer par là sauf les modifications (ou alors garder members comme il est et créer un historic_members avec tout). ¹ https://medium.com/@NyanKiyoshi/how-to-migrate-a-through-to-a-many-to-many-relation-in-django-d09d8bc2c67			

Historique

#1 - 01 octobre 2021 12:56 - Benjamin Dauvergne

- Description mis à jour

#2 - 01 octobre 2021 12:56 - Benjamin Dauvergne

- Description mis à jour

#3 - 01 octobre 2021 15:01 - Paul Marillonnet

- Assigné à mis à Paul Marillonnet

#4 - 19 octobre 2021 11:38 - Paul Marillonnet

- Statut changé de Nouveau à En cours

Commencé à taper un truc dans la branche, erreur cryptique à l'application des migrations, je regarde.

#5 - 21 octobre 2021 15:11 - Paul Marillonnet

Paul Marillonnet a écrit :

Commencé à taper un truc dans la branche, erreur cryptique à l'application des migrations, je regarde.

Le moteur de migration n'arrive pas à appliquer la migration générée.

À l'exécution de "django.db.backends.base.schema.BaseDatabaseSchemaEditor.alter_field", ce bout de code

```
elif old_type is None and new_type is None and (
    old_field.remote_field.through and new_field.remote_field.through and
    old_field.remote_field.through._meta.auto_created and
    new_field.remote_field.through._meta.auto_created):
    return self._alter_many_to_many(model, old_field, new_field, strict)
```

explose parce que new_field.remote_field.through vaut la chaîne 'django_rbac.RoleMember' (i.e. le nom de la classe qui contient les doublets de m2m) et n'a donc pas d'attribut _meta.

Je creuse l'affaire.

#6 - 21 octobre 2021 16:32 - Paul Marillonnet

Paul Marillonnet a écrit :

Je creuse l'affaire.

Bon, en fait il fallait la même mécanique de duplication que pour la classe Role, présente dans django_rbac et a2_rbac. Cette duplication est visible dans la branche.

En revanche, j'ai l'impression que django ne tolère plus la manipulation du billet de blog en description de ce ticket. Ce qui était décrit comme une feature dans le billet semble maintenant être un bug. Je me mange un :

```
$ authentic2-multitenant-manage makemigrations
SystemCheckError: System check identified some issues:

ERRORS:
a2_rbac.Role.members: (fields.E340) The field's intermediary table 'a2_rbac_role_members' clashes with the table name of 'a2_rbac.RoleMember'.
```

sans que django comprenne qu'on cherche précisément à substituer la seconde table à la première.

#7 - 21 octobre 2021 17:36 - Benjamin Dauvergne

Ok, je vais regarder un peu aussi.

#8 - 10 novembre 2021 12:42 - Paul Marillonnet

Bon, je me disais que le souci venait du fait qu'on gère des modèles similaires dans django_rbac et authentic2.a2_rbac. J'ai essayé, par curiosité, de le faire seulement dans a2 mais encore une fois on dirait que django n'est pas prévu pour faire ce genre de chose. Le postulat du billet de blog selon lequel le moteur de migration django n'irait pas altérer la relation m2m me semble faux avec notre version de django : en ajoutant les champs through et through_fields sur le modèle de départ, la relation est forcément altérée même avec l'étape intermédiaire décrite dans le billet de blog.

Je vais voir si on peut passer par une solution à base de migration manuelle du genre :

- création du nouveau modèle intermédiaire m2m
- copie des données de m2m dans le nouveau modèle
- suppression de l'ancienne table
- désignation du nouveau modèle intermédiaire comme celui stockant les Role.members

#9 - 14 février 2022 14:42 - Paul Marillonnet

Je vais y porter un regard frais, en me basant sur ce qui a été fait dans [#57500](#), passé dans la dernière release.

#10 - 17 février 2022 16:37 - Paul Marillonnet

- Statut changé de *En cours* à *Nouveau*

(Bon, toujours rien à proposer qui n'ait pas l'air d'être de la petite bidouille, notamment il y a des comportements du m2m django que je ne comprends pas. il faut d'abord que je lise ce code avant de proposer quelque chose ici.)