

Authentic 2 - Development #42179

IntegrityError: duplicate key value violates unique constraint "a2_rbac_role_admin_scope_ct_id_632506495f341476_u..."

28 avril 2020 09:30 - Sentry Io

Statut:	Fermé	Début:	28 avril 2020
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

<https://sentry.entrouvert.org/entrouvert/publik/issues/1681/>

```
IntegrityError: duplicate key value violates unique constraint "a2_rbac_role_admin_scope_ct_id_632506495f341476_uniq"  
DETAIL:  Key (admin_scope_ct_id, admin_scope_id)=(66, 1898) already exists.  
  
(30 additional frame(s) were not displayed)  
...  
File "django/db/models/sql/compiler.py", line 1112, in execute_sql  
    cursor.execute(sql, params)  
File "django/db/backends/utils.py", line 79, in execute  
    return super(CursorDebugWrapper, self).execute(sql, params)  
File "django/db/backends/utils.py", line 64, in execute  
    return self.cursor.execute(sql, params)  
File "django/db/utils.py", line 94, in __exit__  
    six.reraise(dj_exc_type, dj_exc_value, traceback)  
File "django/db/backends/utils.py", line 64, in execute  
    return self.cursor.execute(sql, params)
```

Révisions associées

Révision 7c4f725b - 23 octobre 2020 10:35 - Paul Marillonnet

a2_rbac: do not break unicity when get-or-creating admin role (#42179)

Révision 89814b51 - 23 octobre 2020 10:35 - Benjamin Dauvergne

misc: fix admin role bad permissions using get_admin_role (#42179)

Révision f7228347 - 23 octobre 2020 10:35 - Benjamin Dauvergne

misc: add caption before external ids in check-and-repair (#42179)

Historique

#1 - 28 avril 2020 09:30 - Lauréline Guérin

- *Projet changé de Suivi des traces à Authentik*

#2 - 28 avril 2020 09:43 - Paul Marillonnet

- *Projet changé de Authentik à Authentic 2*

#3 - 28 avril 2020 10:21 - Frédéric Péters

La "même" lors d'une mise à jour migration plouf,

```
File "/usr/lib/python2.7/dist-packages/django/db/backends/utils.py", line 64, in execute  
    return self.cursor.execute(sql, params)  
  
File "/usr/lib/python2.7/dist-packages/django/db/utils.py", line 94, in __exit__  
    six.reraise(dj_exc_type, dj_exc_value, traceback)
```

```
File "/usr/lib/python2.7/dist-packages/django/db/backends/utils.py", line 64, in execute
```

```
    return self.cursor.execute(sql, params)
```

```
django.db.utils.IntegrityError: duplicate key value violates unique constraint "a2_rbac_role_admin_scope_ct_id_632506495f341476_uniq"
```

```
DETAIL: Key (admin_scope_ct_id, admin_scope_id)=(36, 1014) already exists.
```

#4 - 28 avril 2020 11:00 - Paul Marillonnet

- Fichier 0001-a2_rbac-bugfix-in-administered-object-reference-4217.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

#5 - 28 avril 2020 11:03 - Paul Marillonnet

- Statut changé de Solution proposée à Nouveau

(C'est pas ça me dit-on.)

#6 - 28 avril 2020 11:32 - Paul Marillonnet

À vue de nez après avoir regardé la trace, c'est parce qu'on a unicité sur admin_scope_ct et admin_scope_id alors que le lookup pour le get_or_create lors du

```
assert role.pk == administered_role.get_admin_role().pk
```

se fait sur ces deux attributs mais aussi sur l'OU, qui n'est pas la même pour l'ancien et le nouveau rôle.

#7 - 28 avril 2020 14:05 - Paul Marillonnet

- Fichier 0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch ajouté
- Statut changé de Nouveau à Solution proposée

Parce que je n'ai pas envie d'être celui qui ira corriger à la main quand ça cassera aussi dans 20 jours, cette fois-ci en prod, je verrais bien un garde-fou de ce genre.

#8 - 28 avril 2020 14:21 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Et en fait, ça ne va pas avec ce qu'il y a au dessus dans get_mirror_role, qui laisse croire qu'il y a unicité sur {'admin_scope_ct', 'admin_scope_id', 'ou'} pour les rôles alors qu'en fait c'est {'admin_scope_ct', 'admin_scope_id'}. C'est peut-être cette contrainte d'unicité qu'il faudrait revoir en fait ?

#9 - 28 avril 2020 14:33 - Paul Marillonnet

- Fichier 0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad2.patch ajouté
- Statut changé de En cours à Solution proposée

Paul Marillonnet a écrit :

[...] C'est peut-être cette contrainte d'unicité qu'il faudrait revoir en fait ?

(La version corrigée du patch qui ne demande pas de revoir cette contrainte d'unicité.)

Edit: Et pourtant le kwargument 'ou' de cette méthode get_mirror_role laisse croire que l'unicité n'existe qu'au sein d'une OU. Le doute m'habite.

#10 - 28 avril 2020 16:02 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours
- Assigné à mis à Benjamin Dauvergne

Je m'en occupe merci.

#11 - 29 avril 2020 07:56 - Paul Marillonnet

Ok d'ac.

#12 - 29 avril 2020 09:32 - Benjamin Dauvergne

- Assigné à changé de Benjamin Dauvergne à Paul Marillonnet

On a 6 index d'unicité sur les rôles sans compter la clé primaire :

```
"a2_rbac_role_admin_scope_ct_id_632506495f341476_uniq" UNIQUE CONSTRAINT, btree (admin_scope_ct_id, admin_scope_id)
"a2_rbac_role_unique_idx_0" UNIQUE, btree (ou_id, service_id, slug) WHERE admin_scope_ct_id IS NULL
"a2_rbac_role_unique_idx_1" UNIQUE, btree (service_id, slug) WHERE ou_id IS NULL AND admin_scope_ct_id IS NULL
"a2_rbac_role_unique_idx_2" UNIQUE, btree (ou_id, slug) WHERE admin_scope_ct_id IS NULL AND service_id IS NULL
"a2_rbac_role_unique_idx_3" UNIQUE, btree (slug) WHERE ou_id IS NULL AND admin_scope_ct_id IS NULL AND service_id IS NULL
"a2_rbac_role_uuid_key" UNIQUE CONSTRAINT, btree (uuid)
```

et deux sur permission :

```
"a2_rbac_permission_null_ou_unique_idx_0" UNIQUE, btree (operation_id, ou_id, target_ct_id, target_id)
"a2_rbac_permission_null_ou_unique_idx_1" UNIQUE, btree (operation_id, target_ct_id, target_id) WHERE ou_id IS NULL
```

Pour qu'un `get_or_create()` fonctionne comme attendu il faut que sa clé soit couverte par un index d'unicité et là effectivement ça n'est pas le cas, je valide donc ton analyse.

Mais il y a un autre problème plus haut dans `RoleManager.get_admin_role()`, dans le cas où la cible est un objet scopé (avec un attribut "ou") il ne faut pas inclure l'ou dans la permission car elle est de toute façon ignorée (l'ou ne sert qu'au permission sur des modèles "scopé", i.e. quand `instance(target, ContentType) and target._meta.get_field('ou')` et en cas de changement d'OU on va changer d'objet permission et donc `get_admin_role()` va retourner un rôle différent. Il faudrait différencier ces deux cas.

Comme on sort l'ou de la clé il faudrait par contre avoir un `update_ou=True` dans `mirror_role` et mettre à jour l'OU sur `create=False` (et remplacer `get_or_create` par `update_or_create` ça simplifiera le code).

La migration sur `manage_members` n'était pas parfaite mais ce souci l'a aggravé (à chaque changement d'OU d'un rôle on a eu perte de visibilité et duplication du rôle de gestion :/), c'est le deuxième souci (inclusion de l'OU dans la permission) qui le cause. Le script dans #42190 corrige normalement cela (dédoublonnage des rôles de gestion en conservant les attributions).

#14 - 11 mai 2020 18:58 - Benjamin Dauvergne

Toujours pas; as tu besoin que je relise la branche ou ça va aller ?

#15 - 12 mai 2020 08:54 - Paul Marillonnet

- Fichier `0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch` ajouté

- Statut changé de *En cours* à *Solution proposée*

Benjamin Dauvergne a écrit :

Toujours pas; as tu besoin que je relise la branche ou ça va aller ?

Je n'avais toujours pas eu le temps de prendre en compte tes remarques, mes excuses. C'est chose faite. Je pense qu'il faut aussi relâcher la contrainte d'unicité mentionnée sur les rôles, en l'imposant dans une OU seulement, pas de façon globale.

#16 - 14 mai 2020 08:23 - Benjamin Dauvergne

Ça ne va pas tu as laissé l'ou dans les critères de recherche des rôles :

```
if ou:
    if update_ou:
        update_fields['ou'] = ou
    else:
        kwargs['ou'] = ou <- ici
else:
    kwargs['ou__isnull'] = True <- et là
if create:
    role, created = self.prefetch_related('permissions').get_or_create(
    role, _ = self.prefetch_related('permissions').update_or_create(
```

```
admin_scope_ct=ct,  
admin_scope_id=instance.pk,  
defaults={  
    'name': name,  
    'slug': slug,  
},  
defaults=update_fields,  
**kwargs)
```

si on ne met pas à jour l'ou on ne doit juste pas s'en préoccuper, mais mon avis c'est que pour l'instant on doit toujours mettre à jour l'OU quoi qu'il arrive.

Concernant les permissions il va falloir ne plus y stocker l'OU, tant qu'on y stockera l'OU, sur changement d'OU sur un rôle la permission va changer et l'ancienne devenir orpheline, on ne devrait mettre l'OU sur la permission que si instance est un ContentType et le paramètre ou est défini, pareil ici update_ou n'a pas de sens. Pour récupérer de la situation actuelle il va falloir passer par mon script.

#17 - 14 mai 2020 08:39 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Benjamin Dauvergne a écrit :

si on ne met pas à jour l'ou on ne doit juste pas s'en préoccuper, mais mon avis c'est que pour l'instant on doit toujours mettre à jour l'OU quoi qu'il arrive.

Ok, mais :

- ce n'est pas ce que j'avais compris quand on discutait ici de rajouter explicitement un paramètre update_ou.
- on est bien d'accord que la fonction update_or_create va aller mettre à jour l'objet trouvé en fonction du dictionnaire qu'on lui file en paramètre default, et donc qu'on met à jour l'ou ?
- il me semblait que le code dans son état actuel appelait à relâcher les contraintes d'unicité :

```
unique_together = (  
- ('admin_scope_ct', 'admin_scope_id'),  
+ ('admin_scope_ct', 'admin_scope_id', 'ou'),
```

et donc qu'il faut se préoccuper de l'ou dans tous les cas, sinon c'est s'exposer au risque de MultipleObjectsReturned.

Concernant les permissions il va falloir ne plus y stocker l'OU, tant qu'on y stockera l'OU, sur changement d'OU sur un rôle la permission va changer et l'ancienne devenir orpheline, on ne devrait mettre l'OU sur la permission que si instance est un ContentType et le paramètre ou est défini, pareil ici update_ou n'a pas de sens. Pour récupérer de la situation actuelle il va falloir passer par mon script.

Ah oui, ça me sautait pas aux yeux à la lecture du code, mais je comprends maintenant cette histoire de permission orpheline.

#18 - 14 mai 2020 08:47 - Benjamin Dauvergne

Paul Marillonnet a écrit :

Benjamin Dauvergne a écrit :

si on ne met pas à jour l'ou on ne doit juste pas s'en préoccuper, mais mon avis c'est que pour l'instant on doit toujours mettre à jour l'OU quoi qu'il arrive.

Ok, mais :

- ce n'est pas ce que j'avais compris quand on discutait ici de rajouter explicitement un paramètre update_ou.

En fait on a pas le choix, le rôle de gestion doit suivre son objet; faudrait pas qu'en supprimant l'OU on supprime le rôle de gestion d'un objet dans une autre OU, c'est le fait de rendre ça optionnel qui me chagrine pas de le faire.

- on est bien d'accord que la fonction update_or_create va aller mettre à jour l'objet trouvé en fonction du dictionnaire qu'on lui file en paramètre default, et donc qu'on met à jour l'ou ?

Oui ça c'est ok.

- il me semblait que le code dans son état actuel appelait à relâcher les contraintes d'unicité :
[...] et donc qu'il faut se préoccuper de l'ou dans tous les cas, sinon c'est s'exposer au risque de MultipleObjectsReturned.

C'est pas possible d'avoir plusieurs objets avec le même admin_scope puisque c'est une contrainte.

Concernant les permissions il va falloir ne plus y stocker l'OU, tant qu'on y stockera l'OU, sur changement d'OU sur un rôle la permission va changer et l'ancienne devenir orpheline, on ne devrait mettre l'OU sur la permission que si instance est un ContentType et le paramètre ou est défini, pareil ici update_ou n'a pas de sens. Pour récupérer de la situation actuelle il va falloir passer par mon script.

Ah oui, ça me sautait pas aux yeux à la lecture du code, mais je comprends maintenant cette histoire de permission orpheline.

Faut croiser ça avec la migration post_migrate de Valentin, voir si elle va bien récupérer une permission sans OU (mais j'ai surtout l'impression qu'elle vise le rôle avant la permission, il faudrait faire l'inverse).

#19 - 09 juin 2020 10:17 - Paul Marillonnet

- Fichier 0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch ajouté
- Statut changé de En cours à Information nécessaire
- Patch proposed changé de Oui à Non

Benjamin Dauvergne a écrit :

Faut croiser ça avec la migration post_migrate de Valentin, voir si elle va bien récupérer une permission sans OU (mais j'ai surtout l'impression qu'elle vise le rôle avant la permission, il faudrait faire l'inverse).

J'ai ça en stock qui me semble faire l'affaire hormis le dernier point que tu cites et que je ne comprends pas, à savoir "faire l'inverse [de] vise[r] le rôle avant la permission". Ce qui nous importe ici c'est de corriger tous les rôles d'administration d'autres rôles, non ? Je vois pas comment on peut procéder en partant des permissions pour atterrir sur les rôles et non l'inverse.

#20 - 10 juin 2020 11:07 - Paul Marillonnet

Réponse de Benj sur jabber :

Pour les histoires permissions / rôles, on a un cycle pour les rôles d'administration, le rôle pointe vers la permission via admin_scope_ct et admin_scope_id et la permission pointe sur le rôle via target_ct et target_id.
Donc on peut partir de l'un ou l'autre pour retrouver le couple.
Dès qu'on arrêtera de mettre l'OU sur la permission ça deviendra stable.

#21 - 10 juin 2020 11:08 - Paul Marillonnet

- Statut changé de Information nécessaire à En cours

#22 - 28 juillet 2020 11:59 - Paul Marillonnet

- Patch proposed changé de Non à Oui
- Fichier 0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch ajouté
- Statut changé de En cours à Solution proposée

Zou.

#24 - 15 octobre 2020 12:31 - Benjamin Dauvergne

- Assigné à changé de Paul Marillonnet à Benjamin Dauvergne

#25 - 15 octobre 2020 12:31 - Benjamin Dauvergne

- Fichier 0003-misc-add-caption-before-external-ids-in-check-and-re.patch ajouté
- Fichier 0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch ajouté
- Fichier 0002-misc-fix-admin-role-bad-permissions-using-get_admin_.patch ajouté
- Tracker changé de Bug à Development

Je suis repassé surtout pour ajouter des assertions, qu'on ne reproduise pas le souci à l'avenir et aussi pour faire en sorte que check-and-repair répare le souci.

#26 - 15 octobre 2020 12:32 - Benjamin Dauvergne

- Assigné à changé de Benjamin Dauvergne à Paul Marillonnet

Benjamin Dauvergne a écrit :

Je suis repassé surtout pour ajouter des assertions, qu'on ne reproduise pas le souci à l'avenir et aussi pour faire en sorte que check-and-repair répare le souci.

J'ai testé sur un dump de villeurbanne et tout s'est bien passé, migration + check-and-repair.

#27 - 15 octobre 2020 14:20 - Valentin Deniaud

(mais le test est rouge)

#28 - 15 octobre 2020 14:49 - Benjamin Dauvergne

- Fichier 0003-misc-add-caption-before-external-ids-in-check-and-re.patch ajouté

- Fichier 0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch ajouté

- Fichier 0002-misc-fix-admin-role-bad-permissions-using-get_admin_.patch ajouté

Oublie d'adapter les tests check-and-repair.

#29 - 15 octobre 2020 15:41 - Paul Marillonnet

Pourquoi dans 0002 :

- on passe de count > 1 à count != 2 ? Elle vient d'où cette nouvelle permission d'administration ? Je rate quelque chose, j'ai pas l'impression que cet ajout soit introduit dans 0001 ni dans 0002.
- on a un admin_permission.target.get_admin_role() dans un if qui n'a pas l'air lié à cette instruction (le if vérifie que la permission en elle-même n'est pas dans une OU) ?

#30 - 15 octobre 2020 15:41 - Paul Marillonnet

- Assigné à changé de Paul Marillonnet à Benjamin Dauvergne

#31 - 15 octobre 2020 16:11 - Benjamin Dauvergne

Paul Marillonnet a écrit :

Pourquoi dans 0002 :

- on passe de count > 1 à count != 2 ? Elle vient d'où cette nouvelle permission d'administration ? Je rate quelque chose, j'ai pas l'impression que cet ajout soit introduit dans 0001 ni dans 0002.

Déjà 2 c'est bien supérieur à 1, donc déjà c'est cohérent :) Un rôle d'administration d'un autre rôle a deux permissions d'administration de rôle : une sur le rôle supervisé et une autre sur lui même, ça fait 2, exemple sur villeurbanne :

```
villeurbannetest=# select r.id, r.slug, o.slug as operation_slug, p.target_ct_id, p.target_id from a2_rbac_role as r, a2_rbac_role_permissions as rp, a2_rbac_permission as p, django_rbac_operation o where r.id = rp.role_id and rp.permission_id = p.id and p.operation_id = o.id and r.slug like '%managers-of-role%' order by r.slug;
```

```
id | operation_slug | target_ct_id | target_id
-----+-----+-----+-----
20371 | _a2-managers-of-role-administrateur-fonctionnel | manage_members | 31 | 18602
20371 | _a2-managers-of-role-administrateur-fonctionnel | view | 4 | 5
20371 | _a2-managers-of-role-administrateur-fonctionnel | manage_members | 31 | 20371
19976 | _a2-managers-of-role-agent | manage_members | 31 | 19976
19976 | _a2-managers-of-role-agent | view | 4 | 5
19976 | _a2-managers-of-role-agent | manage_members | 31 | 18604
```

Mais en vrai j'y vais un peu au pif là.

- on a un admin_permission.target.get_admin_role() dans un if qui n'a pas l'air lié à cette instruction (le if vérifie que la permission en elle-même n'est pas dans une OU) ?

J'utilise l'effet de bord qui fait que `get_admin_role()` va virer l'OU de la permission via `.update_or_create(..., defaults={'ou': None})` dans le cas du rôle d'administration d'un rôle; ce qui va corriger le problème.

#32 - 15 octobre 2020 16:32 - Paul Marillonnet

Benjamin Dauvergne a écrit :

Déjà 2 c'est bien supérieur à 1, donc déjà c'est cohérent :) Un rôle d'administration d'un autre rôle a deux permissions d'administration de rôle : une sur le rôle supervisé et une autre sur lui même, ça fait 2, exemple sur villeurbanne :

[...]

Mais en vrai j'y vais un peu au pif là.

Ok, merci c'est plus clair. Simplement je ne comprends pas comment le `elif count > 1: self.warning(...)` n'a pas été à l'origine de faux positifs jusque là à l'exécution de la commande. Je vais regarder.

J'utilise l'effet de bord qui fait que `get_admin_role()` va virer l'OU de la permission via `.update_or_create(..., defaults={'ou': None})` dans le cas du rôle d'administration d'un rôle; ce qui va corriger le problème.

D'ac ok, j'ai pas l'impression que ce comportement correctif soit testé alors je vais relire et dérouler à tête reposée pour qu'on soit sûr de notre coup, et peut-être je vais ajouter un assert dans les tests aussi.

#33 - 15 octobre 2020 17:22 - Benjamin Dauvergne

Paul Marillonnet a écrit :

Benjamin Dauvergne a écrit :

Déjà 2 c'est bien supérieur à 1, donc déjà c'est cohérent :) Un rôle d'administration d'un autre rôle a deux permissions d'administration de rôle : une sur le rôle supervisé et une autre sur lui même, ça fait 2, exemple sur villeurbanne :

[...]

Mais en vrai j'y vais un peu au pif là.

Ok, merci c'est plus clair. Simplement je ne comprends pas comment le `elif count > 1: self.warning(...)` n'a pas été à l'origine de faux positifs jusque là à l'exécution de la commande. Je vais regarder.

Ah si plein mais comme c'est juste des warnings qui ne corrigent rien, ce n'était pas bien grave.

J'utilise l'effet de bord qui fait que `get_admin_role()` va virer l'OU de la permission via `.update_or_create(..., defaults={'ou': None})` dans le cas du rôle d'administration d'un rôle; ce qui va corriger le problème.

D'ac ok, j'ai pas l'impression que ce comportement correctif soit testé alors je vais relire et dérouler à tête reposée pour qu'on soit sûr de notre coup, et peut-être je vais ajouter un assert dans les tests aussi.

Mais tu testes ta modif ou check-and-repair ? Personne utilise check-and-repair à part moi pour l'instant.

#34 - 15 octobre 2020 17:28 - Paul Marillonnet

Benjamin Dauvergne a écrit :

Mais tu testes ta modif ou check-and-repair ? Personne utilise check-and-repair à part moi pour l'instant.

lirc on avait quand même ajouté des tests à cette commande check-and-repair. C'est là que j'aurais imaginé ajouter un assert pour vérifier ce comportement sur l'OU de la permission.

#35 - 15 octobre 2020 17:54 - Benjamin Dauvergne

Paul Marillonnet a écrit :

Benjamin Dauvergne a écrit :

Mais tu testes ta modif ou check-and-repair ? Personne utilise check-and-repair à part moi pour l'instant.

lirc on avait quand même ajouté des tests à cette commande check-and-repair. C'est là que j'aurais imaginé ajouter un assert pour vérifier ce comportement sur l'OU de la permission.

D'ac je vais ajouter ça.

#36 - 19 octobre 2020 10:26 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

#37 - 21 octobre 2020 15:56 - Paul Marillonnet

- Fichier 0004-check-and-repair-test-manage-members-perm-wrong-ou-r.patch ajouté

- Statut changé de En cours à Solution validée

Benjamin Dauvergne a écrit :

D'ac je vais ajouter ça.

On peut faire comme ça. Ack si ça te va d'inclure/squasher ce 0004, que je viens de pousser au bout de la branche.

#38 - 21 octobre 2020 16:00 - Paul Marillonnet

Paul Marillonnet a écrit :

Ack [...]

Par contre le bogue n'est plus critique maintenant que les instances problématiques sont passées à la moulinette du check-and-repair. Je ne verrais pas ça poussé avant vendredi.

#39 - 22 octobre 2020 13:37 - Benjamin Dauvergne

- Fichier 0003-misc-add-caption-before-external-ids-in-check-and-re.patch ajouté

- Fichier 0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch ajouté

- Fichier 0002-misc-fix-admin-role-bad-permissions-using-get_admin_.patch ajouté

Voilà.

PS: comme convenu je pousserai demain.

#40 - 22 octobre 2020 13:59 - Paul Marillonnet

Benjamin Dauvergne a écrit :

Voilà.

PS: comme convenu je pousserai demain.

Top nickel.

#41 - 23 octobre 2020 10:36 - Benjamin Dauvergne

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

```
commit f7228347ce37b9eee39aafb4aa35666911299f21
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Thu Oct 15 12:30:39 2020 +0200
```

misc: add caption before external ids in check-and-repair (#42179)

```
commit 89814b519b198cc67518e7dc4856e5425df64809
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Thu Oct 15 12:30:14 2020 +0200
```

misc: fix admin role bad permissions using get_admin_role (#42179)

```
commit 7c4f725bfc7285bc46ad1a92793bac150f8966a8
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date: Tue Apr 28 14:01:19 2020 +0200
```


a2_rbac: do not break unicity when get-or-creating admin role (#42179)

#42 - 03 novembre 2020 10:16 - Frédéric Péters

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

Fichiers

0001-a2_rbac-bugfix-in-administered-object-reference-4217.patch	990 octets	28 avril 2020	Paul Marillonnet
0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch	3,03 ko	28 avril 2020	Paul Marillonnet
0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch	3,65 ko	28 avril 2020	Paul Marillonnet
0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch	9,22 ko	12 mai 2020	Paul Marillonnet
0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch	7,03 ko	09 juin 2020	Paul Marillonnet
0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch	8,31 ko	28 juillet 2020	Paul Marillonnet
0003-misc-add-caption-before-external-ids-in-check-and-re.patch	1,05 ko	15 octobre 2020	Benjamin Dauvergne
0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch	10,4 ko	15 octobre 2020	Benjamin Dauvergne
0002-misc-fix-admin-role-bad-permissions-using-get_admin_.patch	2,87 ko	15 octobre 2020	Benjamin Dauvergne
0003-misc-add-caption-before-external-ids-in-check-and-re.patch	1,05 ko	15 octobre 2020	Benjamin Dauvergne
0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch	10,4 ko	15 octobre 2020	Benjamin Dauvergne
0002-misc-fix-admin-role-bad-permissions-using-get_admin_.patch	4,47 ko	15 octobre 2020	Benjamin Dauvergne
0004-check-and-repair-test-manage-members-perm-wrong-ou-r.patch	1,63 ko	21 octobre 2020	Paul Marillonnet
0003-misc-add-caption-before-external-ids-in-check-and-re.patch	1,05 ko	22 octobre 2020	Benjamin Dauvergne
0001-a2_rbac-do-not-break-unicity-when-get-or-creating-ad.patch	10,4 ko	22 octobre 2020	Benjamin Dauvergne
0002-misc-fix-admin-role-bad-permissions-using-get_admin_.patch	5,23 ko	22 octobre 2020	Benjamin Dauvergne