

Authentic 2 - Development #65504

auth_oidc : migration 0011 qui casse la clé étrangère des mapping de claim vers le fournisseur (?)

20 mai 2022 11:03 - Paul Marillonnet

Statut:	Fermé	Début:	20 mai 2022
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
0011 : petite manipulation où on supprime l'id et où le baseauthenticator_ptr devient la clé primaire du modèle de fournisseur. Côté OIDCClaimMapping on ne retrouve plus le fournisseur lié.			
Demandes liées:			
Lié à Authentic 2 - Development #53902: avoir une classe de base pour les mod...		Fermé	10 mai 2021

Révisions associées

Révision 3912f264 - 24 mai 2022 11:54 - Valentin Deniaud

authentic2_auth_oidc: attach claims and accounts to new authenticator (#65504)

Révision f5b4bef8 - 24 mai 2022 16:12 - Valentin Deniaud

Revert "authentic2_auth_oidc: attach claims and accounts to new authenticator (#65504)"

This reverts commit 3912f2648c98bafd493cd788b2cf3f57b3f37a4d.

Révision 015da29e - 24 mai 2022 16:45 - Valentin Deniaud

authentic2_auth_oidc: attach claims and accounts to new authenticator (#65504)

Historique

#1 - 20 mai 2022 11:24 - Paul Marillonnet

Et donc j'imagine qu'il manque dans la migration un AlterField sur la FK côté mapping, avant le AlterModelBase défini à l'occasion, quelque chose du genre :

```
diff --git a/src/authentic2_auth_oidc/migrations/0011_auto_20220413_1632.py b/src/authentic2_auth_oidc/migrations/0011_auto_20220413_1632.py
index 713fe5071..b418f7a99 100644
--- a/src/authentic2_auth_oidc/migrations/0011_auto_20220413_1632.py
+++ b/src/authentic2_auth_oidc/migrations/0011_auto_20220413_1632.py
@@ -61,5 +61,16 @@ class Migration(migrations.Migration):
     to='authenticators.BaseAuthenticator',
     ),
     ),
+    migrations.AlterField(
+        model_name='oidcclaimmapping',
+        name='provider',
+        field=models.ForeignKey(
+            related_name='claim_mappings',
+            verbose_name='provider',
+            to='authentic2_auth_oidc.OIDCProvider',
+            to_field='baseauthenticator_ptr',
+            on_delete=models.CASCADE,
+        ),
+    ),
+    AlterModelBase('oidcprovider', 'authenticators.baseauthenticator'),
 ]
diff --git a/src/authentic2_auth_oidc/models.py b/src/authentic2_auth_oidc/models.py
index 944f40270..8b97630fe 100644
--- a/src/authentic2_auth_oidc/models.py
+++ b/src/authentic2_auth_oidc/models.py
@@ -208,7 +208,11 @@ class OIDCClaimMapping(models.Model):
     ]
```

```

provider = models.ForeignKey(
-     to='OIDCProvider', verbose_name=_('provider'), related_name='claim_mappings', on_delete=models.CASCADE
E
+     to='OIDCProvider',
+     to_field='baseauthenticator_ptr',
+     verbose_name=_('provider'),
+     related_name='claim_mappings',
+     on_delete=models.CASCADE,
)
claim = models.CharField(max_length=128, verbose_name=_('claim'), validators=[validate_template])
attribute = models.CharField(max_length=64, verbose_name=_('attribute'))

```

#2 - 20 mai 2022 12:15 - Guillaume Baffoin

- Lié à Development #53902: avoir une classe de base pour les modèles de gestion des moyens d'authentification ajouté

#3 - 20 mai 2022 12:22 - Paul Marillonnet

Et donc effectivement, logiquement après application de 0011, en base la colonne id de la table des fournisseurs oidc a disparu.

La colonne provider_id de la tables oidcclaimmapping pointe vers un id maintenant inexistant.

En l'absence du alterfield(to_field="baseauthenticator_ptr_id") mentionné dans la première note de ce ticket, je ne crois pas que vider la table des mappings pour les récréer à nouveau fonctionne.

Je suis à court d'idées pour trouver une solution (autre qu'un git-revert de cette migration 0011) qui à la fois corrige les confs oidc en erreur, et qui empêche 0011 en l'état actuel de s'appliquer (par exemple lors de la màj des prod de jeudi prochain ☹️) et de casser définitivement les FK existantes.

Si quelqu'un a une idée, je suis preneur.

#4 - 20 mai 2022 12:25 - Pierre Ducroquet

J'ai voulu vérifier si on avait des incohérences en base (test.saas.entrouvert.org) et j'observe plusieurs schémas sur lesquels baseauthenticator_ptr_id n'existe pas. Sur ceux où la migration a été faite, on n'a pour le moment pas d'incohérence visible (on pourrait mettre une FK vers baseauthenticator_ptr_id sans que ça casse).

#5 - 20 mai 2022 12:26 - Pierre Ducroquet

Paul Marillonnet a écrit :

Et donc effectivement, logiquement après application de 0011, en base la colonne id de la table des fournisseurs oidc a disparu.

La colonne provider_id de la tables oidcclaimmapping pointe vers un id maintenant inexistant.

En l'absence du alterfield(to_field="baseauthenticator_ptr_id") mentionné dans la première note de ce ticket, je ne crois pas que vider la table des mappings pour les récréer à nouveau fonctionne.

Il y aurait un problème à ajouter ça ? Ça ne plait pas à django ? Côte SQL ça me choque pas en tout cas.

Je suis à court d'idées pour trouver une solution (autre qu'un git-revert de cette migration 0011) qui à la fois corrige les confs oidc en erreur, et qui empêche 0011 en l'état actuel de s'appliquer (par exemple lors de la màj des prod de jeudi prochain ☹️) et de casser définitivement les FK existantes.

Quelles sont les confs en erreur ?

#6 - 20 mai 2022 12:31 - Paul Marillonnet

Pierre Ducroquet a écrit :

Il y aurait un problème à ajouter ça ? Ça ne plait pas à django ? Côte SQL ça me choque pas en tout cas.

Je notais que ce AlterField aurait dû avoir lieu avant l'opération custom AlterModelBase. Maintenant que c'est fait (et que la colonne id de la table des fournisseur a sauté) je n'ai pas de solution qui va dans ce sens.

Quelles sont les confs en erreur ?

Sur la recette HDS, le fournisseur oidc de l'instance de l'académie de versailles :

https://connexion-ac-versailles.test.entrouvert.org/admin/authentic2_auth_oidc/oidcprovider/7/change/

Sur la recette GL (grandlyon-cut-test-sso-1), le fournisseur oidc (qui il me semble est un autre authentic à part) :

https://moncompte-rec.grandlyon.com/admin/authentic2_auth_oidc/oidcprovider/96/change/

#8 - 20 mai 2022 13:58 - Benjamin Dauvergne

Paul Marillonnet a écrit :

Sur la recette GL (grandlyon-cut-test-sso-1), le fournisseur oidc (qui il me semble est un autre authentic à part) : https://moncompte-rec.grandlyon.com/admin/authentic2_auth_oidc/oidcprovider/96/change/

Sur GLC l'authentification OIDC n'est plus utilisé on peut juste virer ça.

#9 - 20 mai 2022 13:59 - Pierre Ducroquet

Paul Marillonnet a écrit :

Pierre Ducroquet a écrit :

Il y aurait un problème à ajouter ça ? Ça ne plait pas à django ? Côte SQL ça me choque pas en tout cas.

Je notais que ce AlterField aurait dû avoir lieu avant l'opération custom AlterModelBase. Maintenant que c'est fait (et que la colonne id de la table des fournisseur a sauté) je n'ai pas de solution qui va dans ce sens.

Je peux faire un script qui va corriger toutes les bases qui ont subi cette migration pour y ajouter la FK.

Si l'ajout de la FK échoue, ça indiquera qu'il y a une erreur, du coup j'aurai la liste précise.

Comme ça la migration corrigée pourra tourner sur la prod, et le delta avec les autres envs sera minime, modulo les erreurs à corriger.

Ça irait comme ça ?

#10 - 20 mai 2022 14:05 - Benjamin Dauvergne

Rapide investigation sur GLC, ce sont les ids qui ont bougé (excusez moi si je dis des trucs déjà su, j'essaye de comprendre le problème) :

```
In [8]: OIDCProvider.objects.get().pk
Out[8]: 96
```

```
In [9]: set(OIDCClaimMapping.objects.values_list('provider_id', flat=True))
Out[9]: {1}
```

Ici c'est simple y a qu'un provider, un :

```
In [10]: OIDCClaimMapping.objects.update(provider_id=96)
Out[10]: 1
```

m'a suffit à corriger les choses.

Ce qui tombe bien c'est que la contrainte de foreignkey de ClaimMapping.provider_id vers Provider.id a sauté toute seule, je ne sais pas trop ce qui a fait ça. Après correction il faudra éventuellement faire une migration fk -> integer puis integer -> fk pour la rétablir.

PS: fk -> integer risque de planter si il essaye de drop la contrainte qui n'existe déjà plus... il faudra s'en prémunir.

#11 - 20 mai 2022 14:33 - Pierre Ducroquet

Benjamin Dauvergne a écrit :

Ce qui tombe bien c'est que la contrainte de foreignkey de ClaimMapping.provider_id vers Provider.id a sauté toute seule, je ne sais pas trop ce qui a fait ça. Après correction il faudra éventuellement faire une migration fk -> integer puis integer -> fk pour la rétablir.

PS: fk -> integer risque de planter si il essaye de drop la contrainte qui n'existe déjà plus... il faudra s'en prémunir.

Je ne comprends pas trop cette phrase, à moins que ça ne soit un "djangoïsme": une FK est une contrainte attachée à une colonne, donc une migration fk vers autre chose... Ben ça n'a pas trop de sens.

#12 - 20 mai 2022 15:06 - Paul Marillonnet

Benjamin Dauvergne a écrit :

Rapide investigation sur GLC, ce sont les ids qui ont bougé (excusez moi si je dis des trucs déjà su, j'essaye de comprendre le problème) :
[...]

Ici c'est simple y a qu'un provider, un :

[...]

m'a suffit à corriger les choses.

Perso côté AC Versailles j'ai tapé en SQL dans les FK concernées la nouvelle valeur de la clé primaire de l'unique fournisseur. C'est revenu aussi.

Ce qui tombe bien c'est que la contrainte de foreignkey de ClaimMapping.provider_id vers Provider.id a sauté toute seule, je ne sais pas trop ce qui a fait ça.

C'est un [RemoveField de la migration auth_oidc 0011](#) dont je ne comprends pas la nécessité.

#13 - 20 mai 2022 15:12 - Nicolas Roche

Ici c'est simple y a qu'un provider, un :

Sur Nantes en recette il y en a 2 :

https://connexion-nantes-metropole.test.entrouvert.org/admin/authentic2_auth_oidc/oidcprovider/

Étrangement la configuration n'a pas sauté.

#15 - 23 mai 2022 11:12 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

#17 - 23 mai 2022 15:10 - Valentin Deniaud

- Fichier 0001-authentic2_auth_oidc-attach-claims-and-accounts-to-n.patch ajouté

- Tracker changé de Bug à Development

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#18 - 23 mai 2022 15:44 - Valentin Deniaud

- Fichier 0001-authentic2_auth_oidc-attach-claims-and-accounts-to-n.patch ajouté

#19 - 24 mai 2022 10:05 - Serghei Mihai

- Statut changé de Solution proposée à Solution validée

#20 - 24 mai 2022 10:50 - Benjamin Dauvergne

- Statut changé de Solution validée à En cours

Je vais être tatasse, parce que je ne sais pas si le cas se présente vraiment, mais ces lignes:

```
OIDCClaimMapping.objects.filter(provider_id=provider.pk).update(provider_id=base_authenticator.pk)
OIDCAccount.objects.filter(provider_id=provider.pk).update(provider_id=base_authenticator.pk)
```

S'il y a une intersection entre l'ensemble (au sens mathématique) des provider.pk et des base_authenticator.pk ça pourrait ne pas marcher :

- genre changer 2 -> 3
 - puis 3 -> 4
- tout le monde est à 4 à la fin.

Il faudrait moissonner les claimmapping.id et account.id pour chaque provider.pk et ensuite faire l'update ainsi :

```
...
    remap_provider_ids.append((
        OIDCClaimMapping.objects.filter(provider_id=provider.pk).values_list('pk', flat=True),
        OIDCAccount.objects.filter(provider_id=provider.pk).values_list('pk', flat=True),
        base_authenticator.pk,
    ))
for mapping_ids, account_ids, to_provider_id in remap_provider_ids:
    OIDCClaimMapping.objects.filter(id__in=mapping_ids).update(provider_id=to_provider_id)
    OIDCAccount.objects.filter(id__in=account_ids).update(provider_id=to_provider_id)
```

Avec ça je suis ok pour ce soit poussé histoire d'avoir un truc qui passe en prod sans rien péter.

Ensuite j'aimerais bien avoir le fin mot de l'histoire sur cette note de PierreD:

J'ai voulu vérifier si on avait des incohérences en base (test.saas.entrouvert.org) et j'observe plusieurs schémas sur lesquels baseauthenticator_ptr_id n'existe pas. Sur ceux où la migration a été faite, on n'a pour le moment pas d'incohérence visible (on pourrait mettre une FK vers baseauthenticator_ptr_id sans que ça casse).

Est-ce que c'est toujours le cas ? où ? pourquoi ?

Et finalement ça m'embête que la contrainte de FK sur les colonnes provider_id ait sauté et ne revienne pas, à relire les migrations je suppose que c'est un effet de bord du RemoveField(id) (mais j'auré pensé que aurait bloqué le DROP FIELD, visiblement non) parce que Django ne détecte aucune migration nécessaire et je ne vois aucune opération particulière faite sur ces colonnes provider_id (logiquement il aurait fallu les passer en IntegerField comme les clés primaires).

PS: y a aussi moyen de faire ça en un seul update en forgeant une expression Case¹, mais c'est juste plus compliqué pour rien

[1]: <https://docs.djangoproject.com/en/4.0/ref/models/conditional-expressions/#case>

#21 - 24 mai 2022 11:21 - Benjamin Dauvergne

Ok, PierreD me confirme mon intuition, c'est bien le RemoveField('id') qui a supprimé la contrainte de FK, mais le code migration Django ne le sait pas vu qu'on lui a remis une primarykey sous le nez via le AlterModelBase.

Je propose d'ajouter à la migration 0009 en fin de migration deux alterfield sur oidclaimmapping.provider_id et oidccount.provider_id, les transformant en integerfield (en plus c'est nécessaire pour que 0010 marche sinon il va refuser de modifier provider_id sir la contrainte de FK vers provider.id n'est pas respectée), le bonus c'est que Django sait maintenant que la contrainte de FK a disparu.

Ensuite on ajoute un 0012 qui refait l'opération inverse, alterfield de provider_id vers une ForeignKey.

#22 - 24 mai 2022 11:27 - Valentin Deniaud

Benjamin Dauvergne a écrit :

en plus c'est nécessaire pour que 0010 marche sinon il va refuser de modifier provider_id sir la contrainte de FK vers provider.id n'est pas respectée

Yep et le test passe complètement à côté parce que d'abord initialisation de la db avec toutes les migrations dont celle qui fait sauter la contrainte, puis rollback à 0008 mais ça ne restaure pas la contrainte ./

#23 - 24 mai 2022 11:55 - Valentin Deniaud

- Fichier 0001-authentic2_auth_oidc-attach-claims-and-accounts-to-n.patch ajouté

- Statut changé de En cours à Solution proposée

Revoici avec les modifs suggérées

#24 - 24 mai 2022 15:10 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Testé sur un dump de la base Versailles de prod, c'est tout bon.

#25 - 24 mai 2022 15:17 - Valentin Deniaud

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

```
commit 3912f2648c98bafd493cd788b2cf3f57b3f37a4d
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Mon May 23 15:06:23 2022 +0200
```

```
authentic2_auth_oidc: attach claims and accounts to new authenticator (#65504)
```

#26 - 24 mai 2022 16:15 - Valentin Deniaud

- Statut changé de Résolu (à déployer) à En cours

```
commit f5b4bef88c79808fa0c2e4c6916ac10c1c9b6f29
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Tue May 24 16:12:19 2022 +0200
```

```
Revert "authentic2_auth_oidc: attach claims and accounts to new authenticator (#65504)"
```

```
This reverts commit 3912f2648c98bafd493cd788b2cf3f57b3f37a4d.
```

#27 - 24 mai 2022 16:19 - Valentin Deniaud

Ça marche bien seulement sur une nouvelle base, par contre sur une instance déjà migrée 0012 plante parce qu'on a renommé la colonne dans

0009.

#28 - 24 mai 2022 16:46 - Valentin Deniaud

- Fichier 0001-authentic2_auth_oidc-attach-claims-and-accounts-to-n.patch ajouté
- Statut changé de En cours à Solution proposée

Et donc plutôt que de renommer la colonne, dropper explicitement la contrainte devrait avoir l'effet escompté :

```
-         field=models.IntegerField(),
+         field=models.ForeignKey(
+             db_constraint=False,
+             on_delete=django.db.models.deletion.CASCADE,
+             related_name='accounts',
+             to='authentic2_auth_oidc.OIDCProvider',
+             verbose_name='provider',
+         ),
```

Elle est bien ensuite rétablie par le même 0012.

#29 - 24 mai 2022 17:01 - Benjamin Dauvergne

Je vais tester les migrations de master sur versailles puis passer sur la branche, ensuite refaire la même chose directement sur la branche.

#30 - 24 mai 2022 17:05 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

C'est bon.

#31 - 24 mai 2022 17:52 - Valentin Deniaud

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

```
commit 015da29ee465d4714ba3ea4607e6b5f0152d814d
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Mon May 23 15:06:23 2022 +0200
```

```
authentic2_auth_oidc: attach claims and accounts to new authenticator (#65504)
```

#32 - 24 mai 2022 18:14 - Transition automatique

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

#33 - 24 juillet 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-authentic2_auth_oidc-attach-claims-and-accounts-to-n.patch	4,53 ko	23 mai 2022	Valentin Deniaud
0001-authentic2_auth_oidc-attach-claims-and-accounts-to-n.patch	4,71 ko	23 mai 2022	Valentin Deniaud
0001-authentic2_auth_oidc-attach-claims-and-accounts-to-n.patch	7,72 ko	24 mai 2022	Valentin Deniaud
0001-authentic2_auth_oidc-attach-claims-and-accounts-to-n.patch	8,35 ko	24 mai 2022	Valentin Deniaud