

Authentic 2 - Support #45977

Procédure de correction des conflits avec des migrations locales uniquement

20 août 2020 16:03 - Benjamin Renard

Statut:	Nouveau	Début:	20 août 2020
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		
Description			
<p>Sur certaines machines, on a fait des <i>makemigration</i> en local lorsque qu'on avait pas le choix (migrations non présentes dans le paquet et retour arrière impossible). Après coup, on a des conflits de migration qui perdurent après chaque mise à jour et on est obligé de refaire un <i>makemigration</i> à chaque fois pour corrigé les conflits. Quelle est selon vous la bonne méthode pour résoudre une bonne fois pour toutes ces conflits ?</p> <p>J'imaginai éventuellement un export des data de la DB (sans le schéma) puis réinstallation de zéro d'une DB vierge et enfin réimport des données. J'imagine qu'il doit y avoir une table pour le suivi des migrations à exclure de l'export/import, mais ça semble jouable, non ? À moins qu'il y ait plus simple/plus sûre ?</p>			
Demandes liées:			
Lié à Authentic 2 - Development #46984: saml : erreur bytes/str dans x509util...		Fermé	24 septembre 2020

Historique

#1 - 24 septembre 2020 12:05 - Benjamin Renard

Gros problème aujourd'hui lors d'une mise à jour (2.1.27-1~eob80+1 -> 2.67-1~eob90+1): j'avais sur cette installation de vieilles migrations comme expliqué et avec le passage python 2 à python 3, c'est la cata!

Lors de l'upgrade, comme d'habitude, il me dit qu'il y a des migrations manquantes. Je lance donc le *makemigration*, sauf qu'il continue à me dire qu'il manque des migrations. Je relance le *makemigration* et il semble me faire la même migration mais avec un nom de fichier incrémenté. Ça boucle comme ça et plus moyen de lancer le service...

```
(11:18:43) brenard@im.libre-entreprise.com/aconit: root@gavotte-pp~# /usr/bin/authentic2-manage migrate
Operations to perform:
  Apply all migrations: a2_rbac, admin, attribute_aggregator, auth, authentic2, authentic2_auth_fc, authentic2_auth_oidc, authentic2_idp_cas, authentic2_idp_oidc, contenttypes, custom_user, django_rbac, idp, mellon, nonce, saml, sessions
Running migrations:
  No migrations to apply.
  Your models have changes that are not yet reflected in a migration, and so won't be applied.
  Run 'manage.py makemigrations' to make new migrations, and then re-run 'manage.py migrate' to apply them.
root@gavotte-pp~# authentic2-manage makemigrations
Migrations for 'admin':
  /usr/lib/python3/dist-packages/django/contrib/admin/migrations/0008_auto_20200924_1115.py
  - Alter field user on logentry
root@gavotte-pp~# authentic2-manage makemigrations
Migrations for 'admin':
  /usr/lib/python3/dist-packages/django/contrib/admin/migrations/0009_auto_20200924_1116.py
  - Alter field user on logentry
root@gavotte-pp~# authentic2-manage makemigrations --merge
No conflicts detected to merge.
root@gavotte-pp~# /usr/bin/authentic2-manage migrate
Operations to perform:
  Apply all migrations: a2_rbac, admin, attribute_aggregator, auth, authentic2, authentic2_auth_fc, authentic2_auth_oidc, authentic2_idp_cas, authentic2_idp_oidc, contenttypes, custom_user, django_rbac, idp, mellon, nonce, saml, sessions
Running migrations:
  Applying admin.0008_auto_20200924_1115... OK
  Applying admin.0009_auto_20200924_1116... OK
root@gavotte-pp~# /usr/bin/authentic2-manage migrate
Operations to perform:
  Apply all migrations: a2_rbac, admin, attribute_aggregator, auth, authentic2, authentic2_auth_fc, authentic2_auth_oidc, authentic2_idp_cas, authentic2_idp_oidc, contenttypes, custom_user, django_rbac, idp, mellon, nonce, saml, sessions
Running migrations:
```

```
No migrations to apply.
Your models have changes that are not yet reflected in a migration, and so won't be applied.
Run 'manage.py makemigrations' to make new migrations, and then re-run 'manage.py migrate' to apply them.
```

Auriez-vous une idée de me sortir de cette boucle ?

Je me dis que vu la situation, j'aurai dû corriger ce problème de migration avant de lancer la mise à jour. Avant de lancer une restauration complète de la machine (pas de .deb pour faire un rollback sur l'ancienne version...), j'ai donc tenté sur une autre machine la méthode que je visais pour résoudre ce problème, sauf que ça ne semble pas fonctionner:

```
root@sso-dev:~# service authentic2 stop
root@sso-dev:~# authentic2-manage dumpdata --exclude auth.permission --exclude contenttypes > /tmp/authentic.j
son
root@sso-dev:~# su - postgres
postgres@sso-dev:~$ dropdb authentic2
postgres@sso-dev:~$ createdb authentic2
postgres@sso-dev:~$ logout
root@sso-dev:~# authentic2-manage migrate
[...]
root@sso-dev:~# authentic2-manage loaddata /tmp/authentic.json
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/authentic2/serializers.py", line 82, in Deserializer
    for obj in PythonDeserializer(objects, **options):
  File "/usr/lib/python3/dist-packages/django/core/serializers/python.py", line 96, in Deserializer
    for d in object_list:
  File "/usr/lib/python3/dist-packages/authentic2/serializers.py", line 63, in PreDeserializer
    obj = ct.model_class()._default_manager.db_manager(db).get_by_natural_key(*fk_natural_key)
  File "/usr/lib/python3/dist-packages/django/db/models/manager.py", line 85, in manager_method
    return getattr(self.get_queryset(), name)(*args, **kwargs)
  File "/usr/lib/python3/dist-packages/authentic2/managers.py", line 35, in get_by_natural_key
    return self.get(slug=slug)
  File "/usr/lib/python3/dist-packages/django/db/models/query.py", line 380, in get
    self.model._meta.object_name
authentic2.saml.models.DoesNotExist: LibertyProvider matching query does not exist.
```

During handling of the above exception, another exception occurred:

```
Traceback (most recent call last):
  File "/usr/lib/authentic2/manage.py", line 21, in <module>
    execute_from_command_line(sys.argv[:1] + argv)
  File "/usr/lib/python3/dist-packages/django/core/management/__init__.py", line 364, in execute_from_command_
line
    utility.execute()
  File "/usr/lib/python3/dist-packages/django/core/management/__init__.py", line 356, in execute
    self.fetch_command(subcommand).run_from_argv(self.argv)
  File "/usr/lib/python3/dist-packages/django/core/management/base.py", line 283, in run_from_argv
    self.execute(*args, **cmd_options)
  File "/usr/lib/python3/dist-packages/django/core/management/base.py", line 330, in execute
    output = self.handle(*args, **options)
  File "/usr/lib/python3/dist-packages/django/core/management/commands/loaddata.py", line 69, in handle
    self.loaddata(fixture_labels)
  File "/usr/lib/python3/dist-packages/django/core/management/commands/loaddata.py", line 109, in loaddata
    self.load_label(fixture_label)
  File "/usr/lib/python3/dist-packages/django/core/management/commands/loaddata.py", line 166, in load_label
    for obj in objects:
  File "/usr/lib/python3/dist-packages/authentic2/serializers.py", line 88, in Deserializer
    six.reraise(DeserializationError, DeserializationError(e), sys.exc_info()[2])
  File "/usr/lib/python3/dist-packages/django/utils/six.py", line 685, in reraise
    raise value.with_traceback(tb)
  File "/usr/lib/python3/dist-packages/authentic2/serializers.py", line 82, in Deserializer
    for obj in PythonDeserializer(objects, **options):
  File "/usr/lib/python3/dist-packages/django/core/serializers/python.py", line 96, in Deserializer
    for d in object_list:
  File "/usr/lib/python3/dist-packages/authentic2/serializers.py", line 63, in PreDeserializer
    obj = ct.model_class()._default_manager.db_manager(db).get_by_natural_key(*fk_natural_key)
  File "/usr/lib/python3/dist-packages/django/db/models/manager.py", line 85, in manager_method
    return getattr(self.get_queryset(), name)(*args, **kwargs)
  File "/usr/lib/python3/dist-packages/authentic2/managers.py", line 35, in get_by_natural_key
    return self.get(slug=slug)
  File "/usr/lib/python3/dist-packages/django/db/models/query.py", line 380, in get
    self.model._meta.object_name
django.core.serializers.base.DeserializationError: Problem installing fixture '/tmp/authentic.json': LibertyPr
ovider matching query does not exist.
```

Note: Je précise que sur cette machine où j'ai tenté la correction, je suis en version 2.64-1~eob90+1.

Du coup, je sèche... je vois pas bien comment je peux me sortir de cette impasse.

#2 - 24 septembre 2020 13:24 - Paul Marillonnet

Quel est le contenu des fichiers de migration

```
/usr/lib/python3/dist-packages/django/contrib/admin/migrations/0008_auto_20200924_1115.py
```

et

```
/usr/lib/python3/dist-packages/django/contrib/admin/migrations/0009_auto_20200924_1116.py
```

?

#3 - 24 septembre 2020 13:56 - Benjamin Dauvergne

1. Il ne faut jamais créé de nouvelles migrations autre que celles qu'on distribue. Je ne comprends pas quel soucis il y avait avec les migrations manquantes, normalement ça se lance sans problème, on a jamais eu de problème sur nos installations.

2. On ne peut pas dumper une base, migrer, puis recharger le dump, ça ne marche pas, il faut repartir d'une backup complète de la base postgres à partir de là, dumpdata/loaddata ne fonctionne qu'à un même niveau de migration (ce n'est pas du tout un outil de backup, mieux vaut pg_dump pour cela).

#4 - 24 septembre 2020 13:56 - Benjamin Renard

```
root@gavotte-pp~# cat /usr/lib/python3/dist-packages/django/contrib/admin/migrations/0008_auto_20200924_1115.py
```

```
Y
# -*- coding: utf-8 -*-
# Generated by Django 1.11.20 on 2020-09-24 09:15
from __future__ import unicode_literals
```

```
from django.conf import settings
from django.db import migrations, models
import django.db.models.deletion
```

```
class Migration(migrations.Migration):
```

```
    dependencies = [
        ('admin', '0007_auto_20200924_1019'),
    ]

    operations = [
        migrations.AlterField(
            model_name='logentry',
            name='user',
            field=models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to=settings.AUTH_USER_MODEL,
verbose_name='user'),
        ),
    ]
```

```
root@gavotte-pp~# cat /usr/lib/python3/dist-packages/django/contrib/admin/migrations/0009_auto_20200924_1116.py
```

```
Y
# -*- coding: utf-8 -*-
# Generated by Django 1.11.20 on 2020-09-24 09:16
from __future__ import unicode_literals
```

```
from django.conf import settings
from django.db import migrations, models
import django.db.models.deletion
```

```
class Migration(migrations.Migration):
```

```
    dependencies = [
        ('admin', '0008_auto_20200924_1115'),
    ]

    operations = [
        migrations.AlterField(
            model_name='logentry',
            name='user',
            field=models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to=settings.AUTH_USER_MODEL,
verbose_name='user'),
        ),
    ]
```

#5 - 24 septembre 2020 14:04 - Benjamin Renard

Benjamin Dauvergne a écrit :

Il ne faut jamais créé de nouvelles migrations autre que celles qu'on distribue. Je ne comprends quel soucis il y avait avec les migrations manquantes, normalement ça se lance sans problème.

Ça je l'ai bien compris et depuis je ne le fais plus. Mais une fois que la mise à jour est faite et quand on est en prod, il faut bien faire quelques choses si on veut pouvoir relancer le service. C'est pour ça que ce sera utile d'avoir sur vos dépôts Debian les versions précédentes des paquets. Dans cette situation, j'aurai pu faire un rollback sur la version précédente et attendre que vous corrigiez le tir.

En attendant, j'ai une petite dizaine de serveurs dans cette situation et c'était l'objet initial de cette demande: comment corriger le tir sur ces installations pour que les mises à jours à venir s'appliquent sans soucis.

Depuis cette malheureuse expérience, on "impose" systématiquement à nos clients un environnement de preprod pour pouvoir tester les mises à jours dessus et y découvrir les éventuels bugs dans les nouvelles versions que vous livrez. Sans solution de roll-back, c'est encore ce qu'on a trouvé de mieux.

#6 - 24 septembre 2020 14:10 - Benjamin Renard

Benjamin Dauvergne a écrit :

2. On ne peut pas dumper une base, migrer, puis recharger le dump, ça ne marche pas, il faut repartir d'une backup complète de la base postgres à partir de là, dumpdata/loaddata ne fonctionne qu'à un même niveau de migration (ce n'est pas du tout un outil de backup, mieux vaut pg_dump pour cela).

Et bien pourtant, dans l'exemple que je vous ai envoyé, il n'y a pas eu de changement de version, ni de migration sur la DB entre-temps (si on met de côté les anciennes migrations locales qui ont dû être écrasés à l'occasion).

Via un dump/restore PG, je conserve le problème de ces anciennes migrations, d'où ma recherche d'une solution dump/restore des données "utiles" de la DB et non pas celles liées aux mécanismes de suivi des évolutions du modèle de la DB.

Note: nos sauvegardes quotidiennes incluent bien entendu un dump PG systématique de toutes les bases du serveur.

#7 - 24 septembre 2020 14:41 - Benjamin Dauvergne

Pour les migrations inutiles qui ont été appliquées tu peux juste faire un `migrate --fake admin 0007` puis les supprimer du disque, normalement elles n'ont rien fait.

Les migrations normales de la 2.67 se sont-elles appliquées correctement sur la 2.1.27 (je ne discute même pas du fait que c'est une version du 15 mai 2018) ?

Ensuite je n'ai pas de trace ici me disant ce qui ne marche pas quand on lance le serveur. Que se passe-t-il lors d'un `authentic2-manage shell` ou d'un lancement du service, y-a-t-il une trace indiquant une incompatibilité entre le schéma et le code ?

#8 - 24 septembre 2020 16:02 - Benjamin Renard

Benjamin Dauvergne a écrit :

Pour les migrations inutiles qui ont été appliquées tu peux juste faire un `migrate --fake admin 0007` puis les supprimer du disque, normalement elles n'ont rien fait.

Les migrations normales de la 2.67 se sont-elles appliquées correctement sur la 2.1.27 (je ne discute même pas du fait que c'est une version du 15 mai 2018) ?

Ensuite je n'ai pas de trace ici me disant ce qui ne marche pas quand on lance le serveur. Que se passe-t-il lors d'un `authentic2-manage shell` ou d'un lancement du service, y-a-t-il une trace indiquant une incompatibilité entre le schéma et le code ?

Bon alors effectivement, on avance: suite au `migrate --fake`, l'installation du paquet debian s'est bien déroulé suite à un `apt install -f`. Suite à cela, le service ne se lançait pas correctement à cause d'un souci, car `systemd` lançait en `ExecStartPre` un `collectstatic` et la variable `STATIC_ROOT` fournie dans le fichier `authentic.conf` posait problème: je l'ai commenté. J'ai également eu un souci avec le `migrate` lancé en `ExecStartPre` et qui retourne toujours qu'il y a des migrations manquantes: je l'ai également commenté pour contourner. Maintenant, le service démarre... mais ce n'est pas fini: Visiblement, depuis la version `2.64-1~eob90+1` vous avez fait un retour arrière sur la manière de lancer `authentic`: le script `/usr/lib/authentic2/launch-authentic2.sh` n'est plus utilisé et vous appelez `uwsgi` directement. De cette manière, impossible d'accéder à `authentic` à cause du paramètre `ALLOWED_HOSTS` non défini. J'ai tenté de mettre ce paramètre dans `config.py`, mais ensuite c'est l'accès à la DB qui posait problème (nom de DB non-standard) et ensuite je n'aurai pas eu pour autant mon couple de clé/certificat pour les signatures SAML... bref mauvaise piste. J'ai donc modifié le script `systemd` pour réutiliser `launch-authentic2.sh` et là, ça démarre et j'y accède. Sauf que, lorsque j'accède à un service en SAML, j'ai le droit une belle exception:

```
Sep 24 15:50:19 gavotte-pp authentic2[27045]: 104.143.83.241 - f9b77304 ERROR Internal Server Error: /idp/saml2/sso
```

```
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/django/core/handlers/exception.py", line 41, in inner
```

```

        response = get_response(request)
File "/usr/lib/python3/dist-packages/django/core/handlers/base
.py", line 187, in _get_response
        response = self.process_exception_by_middleware(e, request)
File "/usr/lib/python3/dist-packages/django/core/handlers/base
.py", line 185, in _get_response
        response = wrapped_callback(request, *callback_args, **callb
ack_kwargs)
File "/usr/lib/python3.5/contextlib.py", line 30, in inner
        return func(*args, **kwds)
File "/usr/lib/python3/dist-packages/authentic2/decorators.py"
, line 47, in f
        return func(request, *args, **kwargs)
File "/usr/lib/python3/dist-packages/authentic2/decorators.py"
, line 47, in f
        return func(request, *args, **kwargs)
File "/usr/lib/python3/dist-packages/django/views/decorators/c
ache.py", line 57, in _wrapped_view_func
        response = view_func(request, *args, **kwargs)
File "/usr/lib/python3/dist-packages/django/views/decorators/c
srf.py", line 58, in wrapped_view
        return view_func(*args, **kwargs)
File "/usr/lib/python3/dist-packages/authentic2/idp/saml/saml2
_endpoints.py", line 138, in f
        return func(request, *args, **kwargs)
File "/usr/lib/python3/dist-packages/authentic2/idp/saml/saml2
_endpoints.py", line 488, in sso
        server = create_server(request)
File "/usr/lib/python3/dist-packages/authentic2/idp/saml/saml2
_endpoints.py", line 1551, in create_server
        __cached_server = create_saml2_server(request, idp_map=metad
ata_map, options=options)
File "/usr/lib/python3/dist-packages/authentic2/saml/common.py
", line 123, in create_saml2_server
        options=options),
File "/usr/lib/python3/dist-packages/authentic2/saml/common.py
", line 111, in get_saml2_metadata
        return str(metagen)
File "/usr/lib/python3/dist-packages/authentic2/saml/saml2util
s.py", line 282, in __str__
        etree.tostring(self.root_element())
File "/usr/lib/python3/dist-packages/authentic2/saml/saml2util
s.py", line 258, in root_element
        self.generate_services(map, options, self.sso_services)
File "/usr/lib/python3/dist-packages/authentic2/saml/saml2util
s.py", line 202, in generate_services
        self.add_keyinfo(options['key'], None)
File "/usr/lib/python3/dist-packages/authentic2/saml/saml2util
s.py", line 236, in add_keyinfo
        keyinfo(self.tb, key)
File "/usr/lib/python3/dist-packages/authentic2/saml/saml2util
s.py", line 81, in keyinfo
        tb.data(int_to_b64(x509utils.get_rsa_public_key_modulus(key)
))
File "/usr/lib/python3/dist-packages/authentic2/saml/x509utils
.py", line 128, in get_rsa_public_key_modulus
        i = modulus.find('=')
TypeError: a bytes-like object is required, not 'str'

```

Ça ressemble bien à un souci python 2 > 3, peut-être lié au fait que je suis en stretch et donc en python 3.5. Ça vous parle ?

Pour le coup, mon problème de migrations semble presque derrière moi, même si je sais d'avance que je vais avoir des surprises à la prochaine mise à jour. Il faudrait tout de même que j'arrive à trouver une solution pour faire "oublier" ces migrations locales passées sur ces installations.

#9 - 24 septembre 2020 17:43 - Paul Marillonnet

Benjamin Renard a écrit :

Ça ressemble bien à un souci python 2 > 3, peut-être lié au fait que je suis en stretch et donc en python 3.5. Ça vous parle ?

Non ce n'est pas lié à l'utilisation de stretch, mais bien un bogue lors du passage à python 3 de cette partie du code apparemment non testée. Je regarde, c'est dans [#46984](#).

#10 - 24 septembre 2020 20:20 - Benjamin Dauvergne

Le patch est en cours pour le problème pointé.

#11 - 24 septembre 2020 20:20 - Benjamin Dauvergne

- Lié à *Development* #46984: *saml* : erreur bytes/str dans `x509utils.get_rsa_public_key_modulus()` ajouté