Authentic 2 - Development #16514

commande de gestion pour exporter/importer des rôles et OU

26 mai 2017 10:21 - Frédéric Péters

Statut:	Fermé	**Début:**	26 mai 2017
Priorité:	Normal	**Echéance:**	
Assigné à:	Emmanuel Cazenave	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

C'est la partie "rôles" de #12595 (à relire, particulièrement le dernier commentaire de Benjamin), que j'en sors parce qu'elle seule, sans la partie utilisateurs, serait déjà bien utile.

→ avec le même format que les --import-site/--export-site (de passerelle, chrono, combo) pouvoir exporter/importer les rôles (et entités).

On imaginerait alors un paramètre supplémentaire pour la ligne de commande, qui permettrait de limiter les objets qu'on souhaite importer (--import-models entities,roles, genre, mais même ça peut venir plus tard).

Demandes liées:

Lié à Authentic 2 - Development #12595: add a manage command to import roles ...	**Rejeté**	**15 juillet 2016**
Lié à Publik - Development #19852: Export/Import des rôles (et de leur arbore...	**Fermé**	**02 novembre 2017**
Lié à Publik - Development #20770: Pouvoir déployer Publik depuis une UI	**Nouveau**	**18 décembre 2017**
Lié à Hobo - Development #14630: Avoir un import-template générique	**Fermé**	**17 janvier 2017**

Révisions associées

Révision a638275c - 16 avril 2018 11:41 - Benjamin Dauvergne

implement more natural natural keys (#16514)

Révision 17dd1b23 - 16 avril 2018 11:41 - Emmanuel Cazenave

create 'import_site' and 'export_site' commands (#16514)

Historique

#1 - 26 mai 2017 10:21 - Frédéric Péters

- Lié à Development #12595: add a manage command to import roles and users from a data dump ajouté

#2 - 27 septembre 2017 18:11 - Josué Kouka

- Fichier 0001-add-roles-import-export-command-16514.patch ajouté

- Statut changé de Nouveau à En cours

- Assigné à mis à Josué Kouka

- Patch proposed changé de Non à Oui

Un patch de l'import/export de roles dans lequel il manque:

- gestion des relations pour les roles héritant de roles techniques
- une option permettant d'arreter l'import dans lorsque l'OU du roles à importer n'existe pas
- une option permettant d'arreter l'import dans lorsque un des parents du roles à importer n'existe pas

#3 - 27 septembre 2017 18:15 - Josué Kouka

- Fichier 0001-add-roles-import-export-command-16514.patch ajouté

Mauvais patch uploadé

#4 - 27 septembre 2017 18:27 - Frédéric Péters

Au moins,

```
    def export_json(self):
        data = {
            'uuid': self.uuid,
            'slug': self.slug,
            'name': self.name,
        }
```

Ça me semble zapper l'existence d'un attribut description.

Le code de import_json apparait gérer une possibilité de service mais je ne vois pas comment ça peut arriver dans l'export.

#5 - 29 septembre 2017 09:49 - Josué Kouka

- Fichier 0001-add-roles-import-export-command-16514.patch ajouté

Le comportement de l'import dans ce patch.
On s'arrete si:

- l'OU n'est pas trouvé
- le service auquel un role est attaché n'existe pas
- le role parent d'un role n'existe pas seulement si l'option --stop-on-absent-parent est passé.

#6 - 29 septembre 2017 09:58 - Josué Kouka

- Fichier 0001-add-roles-import-export-command-16514.patch ajouté

Fichier de tests oublié.
On s'assure aussi que si le meme fichier importé, aucun doublon n'est crée.

#7 - 29 septembre 2017 10:45 - Thomas Noël

Josué Kouka a écrit :

> Le comportement de l'import dans ce patch.
> On s'arrete si:
>
> - l'OU n'est pas trouvé
> - le service auquel un role est attaché n'existe pas

Mouais, mais hier je t'ai proposé de tester d'abord la présence de toutes les OU et services d'un export, avant de lancer un import. Finalement non ?

- le role parent d'un role n'existe pas seulement si l'option --stop-on-absent-parent est passé.

Pourquoi ça ? Ça voudrait dire qu'on accepte que l'import n'a pas marché et n'a pas su recréer le graphe des relations ?

#8 - 29 septembre 2017 10:46 - Benjamin Dauvergne

Je n'accepterai pas un patch avec Narnia dedans, vraiment.

#9 - 29 septembre 2017 11:10 - Josué Kouka

Thomas Noël a écrit :

> Josué Kouka a écrit :
>
> > Le comportement de l'import dans ce patch.
> > On s'arrete si:
> >
> > - l'OU n'est pas trouvé
> > - le service auquel un role est attaché n'existe pas
>
> Mouais, mais hier je t'ai proposé de tester d'abord la présence de toutes les OU et services d'un export, avant de lancer un import. Finalement non ?

Le bloc d'import est dans le contexte d'une transaction atomique, je pense que ça devrait le faire

- le role parent d'un role n'existe pas seulement si l'option --stop-on-absent-parent est passé.

> Pourquoi ça ? Ça voudrait dire qu'on accepte que l'import n'a pas marché et n'a pas su recréer le graphe des relations ?

T'as raison, on ne devrait pas en fait.

> Je n'accepterai pas un patch avec Narnia dedans, vraiment.

Ok je vais changer (j'étais à cours d'idée)

#10 - 04 octobre 2017 17:17 - Josué Kouka

- Fichier 0001-add-roles-import-export-command-16514.patch ajouté

Avec ce patch, si lors de l'import un OU, Service ou un parent est absent, on sort en erreur.

#11 - 04 octobre 2017 17:19 - Josué Kouka

- Fichier 0001-add-roles-import-export-command-16514.patch ajouté

Suppression d'un import inutile.

#12 - 05 octobre 2017 11:45 - Benjamin Dauvergne

- Assigné à changé de Josué Kouka à Benjamin Dauvergne

Je reprends ce ticket pour faire à ma sauche, notamment je préfèrerai une seule commande d'import/export quitte à ne gérer que les rôles pour l'instant, aussi je tiens au principe d'un objet ImportContext.

#13 - 02 novembre 2017 13:47 - Thomas Noël

- Lié à Development #19852: Export/Import des rôles (et de leur arborescence) ajouté

#14 - 18 décembre 2017 13:05 - Frédéric Péters

- Lié à Development #20770: Pouvoir déployer Publik depuis une UI ajouté

#15 - 19 février 2018 11:09 - Benjamin Dauvergne

- Assigné à changé de Benjamin Dauvergne à Emmanuel Cazenave

#16 - 06 mars 2018 15:12 - Emmanuel Cazenave

- Fichier 0001-import_site-export_site-commands-16514.patch ajouté

Premier essai.

J'ai nommé les commandes import_site et export_site pour l'homogénéité avec les autres briques Publik. gestion des rôles uniquement mais on imagine les utilisateurs pour la suite.
Rien n'est fait sur les permissions de rôle.
Utilisation d'un ImportContext avec pour l'instant aucune option correspondante sur là CLI, je voulais avoir une première revue avant d'aller plus loin.
Il va falloir converger avec #20706 pour la sérialisation des rôles.

#17 - 06 mars 2018 15:21 - Emmanuel Cazenave

Ya quelques trucs pas très propres : une exception non utilisée, la convention de nommage des attributs (avec ou sans underscore) mal utilisée.
J'ai craqué et j'ai besoin d'une review, ce sera corrigé pour le prochain patch.

#18 - 06 mars 2018 16:01 - Benjamin Dauvergne

Emmanuel Cazenave a écrit :

> Il va falloir converger avec #20706 pour la sérialisation des rôles.

Ce n'est pas essentiel.

#19 - 09 mars 2018 10:24 - Emmanuel Cazenave

Vu dans #22377:

Benjamin Dauvergne a écrit :

> Les services n'étant pas unique par slug je demande deux paramètres, le slug de l'OU et le slug du service.

Ici je recherche les services (pour les rattacher aux roles que j'importe), par slug puis par nom, avec dans les deux cas une hypothèse d'unicité, bref un beau bug en perspective.

#20 - 09 mars 2018 10:29 - Benjamin Dauvergne

Par nom c'est clairement pas unique, donc tu peux le faire si effectivement tu trouves un unique exemplaire, sinon abandonne mais pour te simplifier la tâche, utilise l'ou oui.

Par slug et par ou t'es sûr d'en trouver un ou rien.

Dans les précédents patchs on avait dit qu'on exportait pas les rôles attachés à un service (ça doit être écrit quelque part) car ils sont créé automatiquement (il n'y a qu'un seul cas actuellement, les rôles nommés 'Administrateur de <service>' et ils sont créé automatiquement par la commande hobo-deploy d'authentic. Je monte à Paris la semaine prochaine, on pourra voir tout ça.

#21 - 09 mars 2018 10:38 - Emmanuel Cazenave

> Dans les précédents patchs on avait dit qu'on exportait pas les rôles attachés à un service (ça doit être écrit quelque part) car ils sont créé automatiquement (il n'y a qu'un seul cas actuellement, les rôles nommés 'Administrateur de <service>' et ils sont créé automatiquement par la commande hobo-deploy d'authentic. Je monte à Paris la semaine prochaine, on pourra voir tout ça.

J'ai du louper, il y a eu beaucoup d'échanges, bref ça me va bien puisque ça me simplifie la tache. Cool de voir ça en direct avec toi semaine prochaine.

#22 - 16 mars 2018 10:38 - Emmanuel Cazenave

Vu avec Benjamin, je retranscris ses remarques.

Rajouter import/export des OU (en cherchant sur uuid, slug, name).

Stratégie de matching entre roles à importer et roles existant, par ordre de priorité :
1. sur l'uuid
2. sur slug + ou
3. sur nom + ou

Parenté entre roles : cas particulier des parentés entre roles "normaux" et les roles techniques "administrateur du role X", il faut que l'import recrée ces liens de parenté.

Import/export des permissions sur les roles: utiliser les méthodes natural_foreign_key pour retrouver les ContentType (dans la table ContentType, pas de stabilité sur les ids mais stabilité sur le nom des modèles)

Par défaut : supprimer tous les objets liés à des roles et recréer (attributs, permissions).

En option : dry run, supprimer les objets en trop (role et OU présent dans la base mais pas dans l'export) avec une demande de confirmation pas l'utilisateur.

#23 - 16 mars 2018 15:51 - Benjamin Dauvergne

Je valide.

#24 - 16 mars 2018 19:11 - Emmanuel Cazenave

- Sujet changé de commande de gestion pour exporter/importer des rôles à commande de gestion pour exporter/importer des rôles et OU

Je change le titre pour rajouter les OU, puisque qu'on va importer exporter les OU en même temps.

Idée très heureuse d'ailleurs, ça simplifie le code.

#25 - 28 mars 2018 16:18 - Emmanuel Cazenave

- Fichier 0001-create-import_site-and-export_site-commands-16514.patch ajouté

Voilà, ça beaucoup bougé, moins de code, ça marche mieux.

Tests unitaires extensifs, tests manuels en local d'un tenant à un autre concluants (mais il va me falloir clairement de la participation de tout le monde là dessus quand ce sera déployé quelque part)

Ça gère les ou et les roles (parentés, y compris avec les rôles techniques, attributs, permissions), création et update. Dry-run, contexte d'import avec options sur la CLI.

Ce que je n'ai pas fait (et que je n'ai pas envie de faire pour ce premier round, je sature pas mal de ce sujet, et il y aura forcément d'autres rounds).

- suppression des rôles qui sont en base mais pas dans l'import (via l'option role-delete-orphans déjà définie)
- suppression de OU qui sont en base mais pas dans l'import (idem, option ou-delete-orphans)
- recherche des Rôles par nom. La recherche (et donc le match) s'effectue d'abord par uuid, puis par natural_key (combinaison de slug, ou, service).

#26 - 30 mars 2018 10:27 - Frédéric Péters

La partie natural_key, aujourd'hui obligatoire à l'import, crée pour moi quelque chose de trop lié à la structure interne, qui complique terriblement l'utilisation de l'import avec un fichier json produit par ailleurs.

Peut-être qu'il s'agit juste à une série d'endroits d'accepter qu'il n'existe pas de ['natural_key'] et passer à la méthode suivante.

En gros j'aimerais pouvoir donner ça à import_site :

```
{
    "roles": [
        {
            "name": "Debug (42)",
            "slug": "debug-42"
        }
    ]
}
```

Ça passerait avec des modifs comme :

```
@@ -39,11 +39,11 @@ def search_role(role_d):
     Role = get_role_model()
     try:
         return Role.objects.get(uuid=role_d['uuid'])
-    except Role.DoesNotExist:
+    except (Role.DoesNotExist, KeyError):
         pass
     try:
         return Role.objects.get_by_natural_key(*role_d['natural_key'])
-    except Role.DoesNotExist:
+    except (Role.DoesNotExist, KeyError):
         return None
```

et

```
@@ -104,7 +104,10 @@ class RoleDeserializer(object):
             else:
                 self._role_d[key] = value

-        ou_natural_key = self._role_d['natural_key'][1]
+        try:
+            ou_natural_key = self._role_d['natural_key'][1]
+        except KeyError:
+            ou_natural_key = None
         if ou_natural_key:
             self._ou = search_ou(ou_natural_key)
             if self._ou is None:
```

Aussi, je comprends la frilosité mais le comportement par défaut devrait être l'import, et l'option --dry-run, pas l'inverse.

#27 - 30 mars 2018 10:39 - Emmanuel Cazenave

Frédéric Péters a écrit :

> La partie natural_key, aujourd'hui obligatoire à l'import, crée pour moi quelque chose de trop lié à la structure interne, qui complique terriblement l'utilisation de l'import avec un fichier json produit par ailleurs.

Dans l'idée, ils viendraient d'où ces fichiers json ? Un peu de contexte sur les cas d'usage ?

#28 - 30 mars 2018 10:56 - Benjamin Dauvergne

Frédéric Péters a écrit :

> La partie natural_key, aujourd'hui obligatoire à l'import, crée pour moi quelque chose de trop lié à la structure interne, qui complique terriblement l'utilisation de l'import avec un fichier json produit par ailleurs.

On sort un peu du cadre, import/export d'un site identique à l'autre, et on va dériver. Mais ce que je verrai après ce premier ticket ce serait plutôt de pouvoir importer plus des templates de site plutôt que des exports, par exemple une liste de rôle sans OU spécifiée mais à l'import qu'on puisse

mettre --default-ou-slug oullins et on importe une structure basique

> Peut-être qu'il s'agit juste à une série d'endroits d'accepter qu'il n'existe pas de ['natural_key'] et passer à la méthode suivante.
>
> En gros j'aimerais pouvoir donner ça à import_site :
>
> [...]

Non ça je refuse, il faut indiquer une OU (soit explicitement, soit éventuellement implicitement voir commentaire précédent), maintenant ça m'irait de modifier le fonctionnement de Role.natural_key() et RoleManager.get_by_natural_key()

- retourner comme natural_key [role.name, role.slug, ou.natural_key(), role.service and service.natural_key()]
- dans get_by_natural_key() ne rendre que les 3 premiers arguments obligatoires.

Ensuite il faudrait fonctionner comme dumpdata/loaddata qui n'exportent pas la natural_key telle quelle mais la reconstruisent à la volée, en gros ça mange ça:

```
{
   'uuid': 'xxxx',
   'name': 'efefe',
   'slug': 'slug1',
   'ou': {
       'uuid': 'yyyy',
       'name': 'xxxx',
       'slug': 'ou-slug',
   }
   'service': null,
}
```

ça constuit un modèle Role à partir de ça (et donc récursivement ça applique le même principe pour le champ ou), ça appelle .natural_key() et ça tente de retrouver une instance à partir de cela, si trouvé mise à jour sinon création (et pour les champs FK, ou ici, ça ne fait que la recherche bien sûr).

Ma position ce serait donc d'éviter le champ natural_key explicite.

Pour Fred au mieux ça va donner ça:

```
{
   'roles': [
      {
         'name': 'Role1',
         'slug': 'role1',
         'ou': { 'slug': 'default' },
      }
   ]
}
```

ou ce qu'il souhaite si on ajoute une option --default-ou.

> Aussi, je comprends la frilosité mais le comportement par défaut devrait être l'import, et l'option --dry-run, pas l'inverse.

Ça m'irait que par défaut on fasse:

- un dry-run
- on affiche un rapport de pré-import
- on demande une confirmation
- on applique
  et qu'on ait une option --no-input qui passe la confirmation.

Je continue à lire.

#29 - 30 mars 2018 11:00 - Benjamin Dauvergne

Le code de sérialisation de Django (c'est un peu dommage qu'ils aient changé la forme pour l'import par clé naturelle d'un objet ou de ses FK/M2M):

- https://github.com/django/django/blob/master/django/core/serializers/base.py#L268 : FK
- https://github.com/django/django/blob/master/django/core/serializers/base.py#L239 : objet

#30 - 30 mars 2018 11:04 - Frédéric Péters

Par exemple d'un système extérieur qu'on migrerait. Ou d'une version passée ou future d'Authentic qui n'aurait pas exactement la même structure pour natural_key.

Ça vient aussi de mon inquiétude quand on se trouve avec des listes dans le json, plutôt que des dictionnaires, ça rend les évolutions plus compliquées.

Dans mon export local actuel, et c'est un rôle technique donc pas nécessairement représentatif, mais quand même "natural_key":

```
[ "_a2-hobo-superuser", [ "default" ], [ [ "default" ], "portal" ] ],
```

je trouve ça bien moins sûr que

```
"slug": "_a2-hobo-superuser", "ou": {"slug": "default"}, "service": {"slug", "portal", "ou": {"slug": "default
"}}
```

#31 - 30 mars 2018 11:06 - Frédéric Péters

> il faut indiquer une OU

Ok mais comme il y a déjà une OU indiquée comme OU par défaut au niveau du site, je me disais que l'information pouvait être utilisée.

#32 - 30 mars 2018 15:04 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Par exemple d'un système extérieur qu'on migrerait. Ou d'une version passée ou future d'Authentic qui n'aurait pas exactement la même structure pour natural_key.
>
> Ça vient aussi de mon inquiétude quand on se trouve avec des listes dans le json, plutôt que des dictionnaires, ça rend les évolutions plus compliquées.
>
> Dans mon export local actuel, et c'est un rôle technique donc pas nécessairement représentatif, mais quand même "natural_key":
>
> ["_a2-hobo-superuser", ["default"], [["default"], "portal"]],
>
> je trouve ça bien moins sûr que
>
> "slug": "_a2-hobo-superuser", "ou": {"slug": "default"}, "service": {"slug", "portal", "ou": {"slug": "default"}}

Pour ça on est d'accord.

#33 - 30 mars 2018 15:08 - Benjamin Dauvergne

Frédéric Péters a écrit :

> > il faut indiquer une OU
>
> Ok mais comme il y a déjà une OU indiquée comme OU par défaut au niveau du site, je me disais que l'information pouvait être utilisée.

Je préfère qu'on évite l'implicite, dans le cas général d'import entre site mono-collectivité, on aura toujour ou_slug=='default' et ça marchera tout seul d'export en import et à la génération depuis un système tiers (ce que vraiment on a pas prévu de faire là maintenant et qu'on ne fait jamais pour l'instant) on pensera à mettre 'ou': {'slug': 'default'}, si on se retrouve sur un site où le slug default n'existe pas ça plantera et ça nous obligera à réfléchir à ce qu'on est en train de faire.

À la rigueur, je veux bien qu'on accepte l'absence d'ou mais uniquement si le site où on charge n'en a qu'une.

#34 - 03 avril 2018 10:42 - Emmanuel Cazenave

- Fichier 0001-create-import_site-and-export_site-commands-16514.patch ajouté

Nouveau patch.

J'ai changé le format de sérialisation : exit la natural key et vive l'explicite à la "slug": "_a2-hobo-superuser", "ou": {"slug": "default"}, "service": {"slug", "portal", "ou": {"slug": "default"}}
Mais je me sers toujours de la natural key pour matcher, que je recalcule maintenant à la volée.
Le calcul est bien encapsulé dans une fonction, ça sera facile de choisir une autre fonction dynamiquement en tenant compte d'une nouvelle option de CLI par exemple, pour des comportements d'import plus subtils.

Par défaut la commande fait maintenant l'import, avec un prompt de confirmation, désactivable en passant --yes.
Et du coup le dry run s'obtient en passant --dry-run.

On est bon comme ça mes petits lapins ?

#35 - 03 avril 2018 10:47 - Frédéric Péters

> Par défaut la commande fait maintenant l'import, avec un prompt de confirmation, désactivable en passant --yes.

Plutôt --noinput pour cette option, pour suivre ce qui est fait pour migrate et collectstatic ? (même si dans les autres modules on n'a pas ce genre d'option et qu'au fond je m'en passerais bien ici aussi, plutôt que devoir aller le mettre partout).

#36 - 03 avril 2018 10:57 - Emmanuel Cazenave

Oui, parce que moi aussi j'apprécie les vertus de la régularité.
Et je veux bien la virer aussi, pour peu qu'on me fasse pas la remettre si je la vire : Benjamin c'est bon pour toi ?

#37 - 03 avril 2018 11:19 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Par défaut la commande fait maintenant l'import, avec un prompt de confirmation, désactivable en passant --yes.
>
> Plutôt --noinput pour cette option, pour suivre ce qui est fait pour migrate et collectstatic ? (même si dans les autres modules on n'a pas ce genre d'option et qu'au fond je m'en passerais bien ici aussi, plutôt que devoir aller le mettre partout).

Mais tu imagines déjà des flopées d'import/export automatiques dans tous les sens ou bien ?

#38 - 03 avril 2018 11:35 - Frédéric Péters

Lors des déploiements, qu'hobo ait connaissance de modèles et provoque un import_site whatever pour toutes les applications; là-dessus donc, plutôt qu'avoir à coder dans hobo que authentic est différent et demande un flag --noinput, je préfère un comportement homogène (mais si ça doit passer par l'ajout d'une confirmation et de la prise en charge d'un --noinput dans les autres applications, so be it).

#39 - 03 avril 2018 11:37 - Benjamin Dauvergne

```
#  Borrowed from https://bugs.python.org/issue10049#msg118599
@contextlib.contextmanager
def provision_contextm(dry_run):
    multitenant = False
    try:
        import hobo.agent.authentic2
        multitenant = True
    except ImportError:
        pass
    if dry_run and multitenant:
        with hobo.agent.authentic2.provisionning.Provisionning():
            yield
    else:
        yield
```

un peu alambiqué, fais juste un test sur la présence de 'hobo.agent.authentic2' dans INSTALLED_APPS (la situation n'est pas vraiment multitenant ou pas, mais provisionning ou pas, ici malheureusement par le truchement de la Publik-isation d'authentic tout ça est un peu mélangé).

La duplication du code de calcul des natural key n'est pas super DRY, normalement ça irait tout seul si:
1. tu résoud la référence à l'OU, (si on ne trouve pas, boum)
2. puis tu copies la sérialisation et tu remplaces le champ 'ou' par la valeur obtenue en 1,
3. ensuite tu peux directement passer ta sérialisation au constructeur de Role,
4. enfin tu peux appeler la méthode natural_key() sur l'objet temporaire créé en 3 (c'est à peux près ce que fait le code de dé-sérialisation de Django)

Finalement la reprise du code d'export de passerelle ne sert un peu à rien:

- contrairement à Passerelle on a pas encore la visée d'exporter tout et n'importe quoi
- le code n'est pas factorisé ce qui est tout l'intérêt dans passerelle

Je remplacerai volontiers ça

```
        concrete_fields = [f for f in self.__class__._meta.get_fields()
                           if f.concrete and not f.is_relation]
        for field in concrete_fields:
            if field.name == 'id':
                continue
            value = getattr(self, field.attname)
            if isinstance(field, SIMPLE_SERIALIZABLE_FIELDS):
                d[field.name] = value
```

```
            else:
                raise Exception('export_json: field %s of ressource class %s is unsupported' % (
                    field, self.__class__))
```

dans Role.export_json() par ça:

```
d['uuid'] = self.uuid
d['slug'] = self.slug
d['name'] = self.name
```

et ça:

```
        d['ou'] = None
        if self.ou:
            d['ou'] = {'uuid': self.ou.uuid, 'slug': self.ou.slug, 'name': self.ou.name}
```

par un simple

```
d['ou'] = self.ou and self.ou.export_json(full=False)  # full enable exporting other attributes uuid, slug and
 name
```

idem pour Service (et comme on en est pas encore à exporter correctement les services, full=True doit raiser NotImplementedError sur Service).

Et je ferai la même chose pour Permission (on exporter le truc correctement et on cacul les natural_key à la volée, avec un export_json()/import_json() sur l'objet Permission).

#40 - 03 avril 2018 11:37 - Frédéric Péters

- Lié à Development #14630: Avoir un import-template générique ajouté

#41 - 03 avril 2018 11:38 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Lors des déploiements, qu'hobo ait connaissance de modèles et provoque un import_site whatever pour toutes les applications; là-dessus donc, plutôt qu'avoir à coder dans hobo que authentic est différent et demande un flag --noinput, je préfère un comportement homogène (mais si ça doit passer par l'ajout d'une confirmation et de la prise en charge d'un --noinput dans les autres applications, so be it).

Bon ok alors, on garde la confirmation que si --dry-run est invoqué, et par défaut c'est non.

#42 - 03 avril 2018 11:41 - Frédéric Péters

> Bon ok alors, on garde la confirmation que si --dry-run est invoqué, et par défaut c'est non.

La demande de confirmation sur un tir à blanc ce n'est vraiment pas utile, je trouve. Ça me va de garder --noinput ici et quand il y aura des appels depuis hobo, hobo gérera.

#43 - 03 avril 2018 11:54 - Emmanuel Cazenave

Finalement la reprise du code d'export de passerelle ne sert un peu à rien:

- contrairement à Passerelle on a pas encore la visée d'exporter tout et n'importe quoi
- le code n'est pas factorisé ce qui est tout l'intérêt dans passerelle

Mais tu disais exactement l'inverse ici : https://dev.entrouvert.org/issues/12595#note-32 !!! Et je trouve que tu avais raison.
Effectivement ça manque de factorisation, je veux bien faire un effort là dessus.
Mais me faire revenir en arrière alors que j'ai fait l'effort de respecter tes remarques....

#44 - 03 avril 2018 12:15 - Benjamin Dauvergne

Emmanuel Cazenave a écrit :

> Finalement la reprise du code d'export de passerelle ne sert un peu à rien:
>
> - contrairement à Passerelle on a pas encore la visée d'exporter tout et n'importe quoi
> - le code n'est pas factorisé ce qui est tout l'intérêt dans passerelle

> Mais tu disais exactement l'inverse ici : https://dev.entrouvert.org/issues/12595#note-32 !!! Et je trouve que tu avais raison.
> Effectivement ça manque de factorisation, je veux bien faire un effort là dessus.
> Mais me faire revenir en arrière alors que j'ai fait l'effort de respecter tes remarques....

Le problème c'est que ce n'est pas factorisé, j'aurai du être plus explicite mais l'idée d'un tel code c'est d'avoir une base pour aller vers un export de tous les modèles, là tel que c'est fait ça ne nous avancera pas sur cette voie, donc soit tu factorises soit tu simplifies (j'en ai déjà jeter plus que ça du code tu ne me feras pas pleurer :-)).

Au passage si ça peut gérer directement les FK et M2M via natural_key, +100 (aussi pour te simplifier la vie, ce code générique devrait prendre en paramètre optionnel la liste des champs qu'on souhaite exporter).

#45 - 03 avril 2018 14:10 - Emmanuel Cazenave

En l'état mon patch est loin du niveau de généricité de passerelle mais permets quand même de pas avoir à toucher au code d'import/export à chaque fois qu'on rajoute un champ 'simple' à OU ou Role, c'était ça mon objectif.

#46 - 03 avril 2018 16:12 - Benjamin Dauvergne

Emmanuel Cazenave a écrit :

> En l'état mon patch est loin du niveau de généricité de passerelle mais permets quand même de pas avoir à toucher au code d'import/export à chaque fois qu'on rajoute un champ 'simple' à OU ou Role, c'était ça mon objectif.

Si tu peux simplement faire une fonction, generic_export_json(instance, fields=()) ce sera un début de factorisation, prends tous les champs si fields est vide, sinon uniquement ceux dans fields.

#47 - 03 avril 2018 20:00 - Emmanuel Cazenave

J'ai poussé dans wip/import-export-role

> un peu alambiqué, fais juste un test sur la présence de 'hobo.agent.authentic2' dans INSTALLED_APPS (la situation n'est pas vraiment multitenant ou pas, mais provisionning ou pas, ici malheureusement par le truchement de la Publik-isation d'authentic tout ça est un peu mélangé).

Done.

> La duplication du code de calcul des natural key n'est pas super DRY, normalement ça irait tout seul si:

....

Tu parles de build_role_natural_key ? Si on s'interdit d'utiliser natural_key dans la sérialisation, il faut bien que je la recalcule quand je déserialise pour pouvoir appeler get_by_natural_key. Donc ta remarque en fait c'est : ne jamais de servir de get_by_natural_key. Mais pourquoi donc ? Ça me mâche le travail et c'est fait pour ça il me semble non ?

> Et je ferai la même chose pour Permission (on exporter le truc correctement et on calcule les natural_key à la volée, avec un export_json()/import_json() sur l'objet Permission).

Sûr de sûr ? Là j'ai fait un natural_key d'un coté, un get_by_natural_key de l'autre et ça a marché direct. Le modèle étant un peu compliqué avec entre autres sa généric relation, ça va faire pas mal de code pour arriver au même résultat. Le jeu en vaut-il la chandelle ? Pitié.

> Finalement la reprise du code d'export de passerelle ne sert un peu à rien:

J'ai tout viré et opté pour ta solution simple.

#48 - 04 avril 2018 18:44 - Emmanuel Cazenave

- Fichier 0001-create-import_site-and-export_site-commands-16514.patch ajouté

- Fichier export_site.json ajouté

Nouveau patch, j'ai poursuivi l'effort avec un export_json pour les permissions, et un gros test grandeur nature.

Je joins ici export_site.json, export brut.

#49 - 05 avril 2018 10:15 - Benjamin Dauvergne

- Fichier 0001-implement-more-natural-natural-keys-16514.patch ajouté

On a encore des natural_key à l'ancienne pour les permissions au niveau de target_ct et target.

Aussi sur les rôles admin_scope_id et admin_scope_ct_id sont des PK il faut mettre des clés naturelles, si possible avec le format explicite qu'on s'est donné (et que Django aurait du adopter, si ce n'est que leur implémentation des clés naturelles est pourrave).

Je pose ma contribution au problème, des clés vraiment naturelles:

- nouvelle méthode pour tous les objets nommées Model.natural_key_json()
- nouvelle méthode pour tous les managers nommées Manager.get_by_natural_key_json(d)

Pour l'utiliser il faut poser Model._meta.natural_key = [[première liste de clés naturelles], [deuxième], etc..], on peut mettre une seule liste directement si ça suffit ou une liste de listes.

Ça devrait t'aider.

#50 - 09 avril 2018 14:21 - Emmanuel Cazenave

- Fichier 0002-add-natural_key_json-to-Service-16514.patch ajouté

- Fichier 0003-create-import_site-and-export_site-commands-16514.patch ajouté

- Fichier export_site_natural_key_json.json ajouté

0002-add-natural_key_json-to-Service-16514.patch finit le travail de Benjamin

0003-create-import_site-and-export_site-commands-16514.patch les commandes d'import/export basées sur ce travail.

export_site_natural_key_json.json nouvel exemple d'export

J'ai enlevé le prompt de confirmation à l'import (parce qu'on peut aussi satisfaire Frédéric, et qu'on a déjà le dry-run pour se rassurer).

J'ai enlevé l'énorme immonde test "grandeur nature" (qui n'était qu'une flemme de ma part d'écrire des tests unitaires appropriés sur les permissions, ce qui est maintenant fait).

#51 - 12 avril 2018 12:16 - Benjamin Dauvergne

- Tu peux merger le 0001 et le 0002.
- On a oublié un truc dont on a avait parlé au tableau: exporter les fils des rôles d'administration des rôles, dans Role.export_json():

```
if admin_roles:
    admin_role = self.get_admin_role()
    for relation in admin_role.child_relation.all():
       d.setdefault('admin_roles', []).append(relation.child.natural_key_json())
```

idem à la désérialisation

#52 - 13 avril 2018 16:44 - Emmanuel Cazenave

- Fichier 0001-implement-more-natural-natural-keys-16514.patch ajouté

- Fichier 0002-create-import_site-and-export_site-commands-16514.patch ajouté

Vu avec Benjamin.
En fait pas besoin d'avoir un traitement particulier pour les fils de rôle d'administration des rôles.
Dans la première passe de l'import on ne déserialise pas les rôles techniques, mais pour chaque rôle déserialisé on appelle get_admin_role, ce qui garantit qu'on disposera des tous les rôles nécessaires pour mener à bien la création des liens de parenté, dans une deuxième passe.

J'ai mis un commentaire dans le code à ce sujet dans 0002-create-import_site-and-export_site-commands-16514.patch, le code n'a pas bougé.

Et 0001-implement-more-natural-natural-keys-16514.patch, un seul patch pour le nouveau système de natural_key.

#53 - 13 avril 2018 17:18 - Emmanuel Cazenave

- Fichier 0002-create-import_site-and-export_site-commands-16514.patch ajouté

oublié de supprimer un fichier data de tests non utilisé, et en plus pas à jour, donc nouveau patch.

#54 - 16 avril 2018 10:52 - Benjamin Dauvergne

Ack.

#55 - 16 avril 2018 11:48 - Emmanuel Cazenave

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

```
commit a638275c0999d9f0ef7944b03b45433a8c2e24cc
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Thu Apr 5 10:10:11 2018 +0200

    implement more natural natural keys (#16514)

commit 17dd1b2338202633f88287be080133193542de5a
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date:   Mon Apr 9 14:10:21 2018 +0200

    create 'import_site' and 'export_site' commands (#16514)
```

#56 - 30 novembre 2018 12:29 - Emmanuel Cazenave

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

#57 - 13 décembre 2018 22:29 - Benjamin Dauvergne

- Statut changé de Solution déployée à Fermé

Fichiers

0001-add-roles-import-export-command-16514.patch	7,02 ko	27 septembre 2017	Josué Kouka
0001-add-roles-import-export-command-16514.patch	7,17 ko	27 septembre 2017	Josué Kouka
0001-add-roles-import-export-command-16514.patch	9,51 ko	29 septembre 2017	Josué Kouka
0001-add-roles-import-export-command-16514.patch	21,1 ko	29 septembre 2017	Josué Kouka
0001-add-roles-import-export-command-16514.patch	23,2 ko	04 octobre 2017	Josué Kouka
0001-add-roles-import-export-command-16514.patch	23,2 ko	04 octobre 2017	Josué Kouka
0001-import_site-export_site-commands-16514.patch	35,9 ko	06 mars 2018	Emmanuel Cazenave
0001-create-import_site-and-export_site-commands-16514.patch	45,1 ko	28 mars 2018	Emmanuel Cazenave
0001-create-import_site-and-export_site-commands-16514.patch	48,8 ko	03 avril 2018	Emmanuel Cazenave
0001-create-import_site-and-export_site-commands-16514.patch	96,5 ko	04 avril 2018	Emmanuel Cazenave
export_site.json	45,9 ko	04 avril 2018	Emmanuel Cazenave
0001-implement-more-natural-natural-keys-16514.patch	8,48 ko	05 avril 2018	Benjamin Dauvergne
0002-add-natural_key_json-to-Service-16514.patch	2,68 ko	09 avril 2018	Emmanuel Cazenave
0003-create-import_site-and-export_site-commands-16514.patch	95 ko	09 avril 2018	Emmanuel Cazenave
export_site_natural_key_json.json	46,2 ko	09 avril 2018	Emmanuel Cazenave
0001-implement-more-natural-natural-keys-16514.patch	9,19 ko	13 avril 2018	Emmanuel Cazenave
0002-create-import_site-and-export_site-commands-16514.patch	95,1 ko	13 avril 2018	Emmanuel Cazenave
0002-create-import_site-and-export_site-commands-16514.patch	47,5 ko	13 avril 2018	Emmanuel Cazenave