

Passerelle - Development #13887

possibilité d'export/import de paramétrage

07 novembre 2016 12:47 - Frédéric Péters

Statut:	Fermé	Début:	07 novembre 2016
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	100%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
On a maintenant des connecteurs qui deviennent parfois longs à configurer (<hint>connecteur CSV</hint>), du coup ça serait bien d'avoir une procédure d'export/import.			
Demandes liées:			
Lié à Hobo - Development #14630: Avoir un import-template générique		Fermé	17 janvier 2017

Révisions associées

Révision ee86d7ed - 03 mars 2017 16:11 - Benjamin Dauvergne

create unique index on ApiUser.username (#13887)

Révision 9a4794fd - 03 mars 2017 16:11 - Benjamin Dauvergne

create unique index on AccessRight.* (#13887)

It will guarantee that get_or_create() calls in the BaseResource.import_json() function will always work.

Révision 6d3782a0 - 06 mars 2017 13:11 - Benjamin Dauvergne

implement JSON import/export (fixes #13887)

You can import/export individual resources or full sites.

```
passerelle-manage export_site >export.json
```

```
passerelle-manage import_site [--import-users] [--clean] [--overwrite] [--if-empty] export.json
```

Historique

#1 - 07 novembre 2016 12:48 - Frédéric Péters

- Description mis à jour

#2 - 16 janvier 2017 16:51 - Benjamin Dauvergne

Ébauche rapide pour l'export:

- partir des sérialisation classiques Django pour les modèles Ressource et APIUser, virer les fioriture Django et ne garder qu'un champ 'resource_type': ...
- concernant ressource on remplace les champs fichiers avec un export base64 du fichier en essayant de conserver un tout petit son nom (un bête basename sur le chemin)
- on ajoute un une liste d'objet AccessRight ans les sérialisations des ressources, on point sur les APIUser via leur username

Pour l'import:

- on recrée les APIUser avec get_or_create() en utilisant username comme clé (modulo que username n'a pas le flag unique=True)
- on recrée tous les fichiers pour chaque ressource, on remplace la sérialisation par le chemin obtenu
- on recrée les objets à partir de leur slug .get_or_create() au passage on recrée les fichiers
- on recrée les AccessRights à partir des id généré à l'étape précédente et des username

#3 - 16 janvier 2017 16:52 - Benjamin Dauvergne

Vérifier l'existence de username en doublon en prod et recette et sinon corriger ça, puis ajouter unique à APIUser.username.

#4 - 16 janvier 2017 17:46 - Frédéric Péters

À noter aussi que ça sera utile d'avoir la possibilité via l'UI d'exporter/importer un connecteur particulier. (à l'écriture du ticket je pensais surtout à ça).

#5 - 16 janvier 2017 18:34 - Benjamin Dauvergne

À priori j'aurais trois nouvelles méthodes au niveau des ressources:

- `export_json()`, à surcharger par chaque ressource pour traitement spécifique (exporter des objets liés en dehors des `AccessRight` gérés par `BaseResource`), peut faire un `raise NotImplementedError` pour indiquer qu'on ne souhaite pas exporter cette ressource (pour les vieux trucs)
- `import_json()` méthode publique au niveau de `BaseResource`, recherche la bonne classe de Ressource avant d'appeler `import_json_impl()` dessus
- `import_json_impl()` au niveau de chaque Ressource avec une implémentation basique dans `BaseResource`

J'ai fait la liste des ressources ou autres modèles qui y sont liés:

```
passerelle/apps/base_adresse/models.py:class BaseAdresse(BaseResource):
passerelle/apps/base_adresse/models.py:class StreetModel(models.Model):
```

Dois-je exporter `StreetModel` ? Je dirai non a priori, ça va se recharger tout seul.

```
passerelle/apps/choosit/models.py:class ChoositRegisterNewsletter(models.Model):
passerelle/apps/choosit/models.py:class ChoositRegisterGateway(BaseResource):
```

On n'exporte pas, il faudrait récupérer les Newsletter et je pense qu'on s'en fout.

```
passerelle/apps/clicrdv/models.py:class ClicRdv(BaseResource):
passerelle/apps/concerto/models.py:class Concerto(BaseDataSource):
passerelle/apps/csvdatasource/models.py:class Query(models.Model):
passerelle/apps/csvdatasource/models.py:class CsvDataSource(BaseResource):
```

La classe qui nous intéresse vraiment, exporter fichier et requêtes.

```
passerelle/apps/family/models.py:class GenericFamily(BaseResource):
passerelle/apps/family/models.py:class FamilyLink(models.Model):
passerelle/apps/family/models.py:class Family(models.Model):
passerelle/apps/family/models.py:class Person(models.Model):
passerelle/apps/family/models.py:class Adult(Person):
passerelle/apps/family/models.py:class Child(Person):
passerelle/apps/family/models.py:class Invoice(models.Model):
```

Je dirai qu'on exporte pas les données.

```
passerelle/apps/gdc/models.py:class Gdc(BaseResource):
passerelle/apps/makorepost/models.py:class BaseMakoRepost(BaseRepost):
passerelle/apps/makorepost/models.py:class MakoPrototypeRepost(BaseMakoRepost):
passerelle/apps/makorepost/models.py:class MakoRepost(BaseMakoRepost):
```

Ce n'est même pas une ressource, on ignore.

```
passerelle/apps/mobyts/models.py:class MobytsSMSSGateway(BaseResource, SMSSGatewayMixin):
passerelle/apps/orange/models.py:class OrangeSMSSGateway(BaseResource, SMSSGatewayMixin):
passerelle/apps/ovh/models.py:class OVHSMSSGateway(BaseResource, SMSSGatewayMixin):
passerelle/apps/oxyd/models.py:class OxydSMSSGateway(BaseResource, SMSSGatewayMixin):
passerelle/apps/pastell/models.py:class Pastell(BaseResource):
passerelle/apps/solis/models.py:class SolisReferential(BaseDataSource):
passerelle/apps/solis/models.py:class SolisListeCommunes(BaseDataSource):
passerelle/apps/solis/models.py:class SolisListeLieux(BaseDataSource):
passerelle/apps/solis/models.py:class SolisListeHomonymes(BaseDataSource):
passerelle/apps/sqlrepost/models.py:class SQLRepost(BaseRepost):
passerelle/apps/sqlrepost/models.py:class SQLStatement(models.Model):
passerelle/apps/sqlrepost/models.py:class SQLBinding(models.Model):
```

Idem que `MakoRepost`.

```
passerelle/base/models.py:class ApiUser(models.Model):
passerelle/base/models.py:class TemplateVar(models.Model):
passerelle/base/models.py:class InheritanceManager(ModelUtilsInheritanceManager):
passerelle/base/models.py:class BaseResource(models.Model):
passerelle/base/models.py:class AccessRight(models.Model):
passerelle/contrib/agoraplus/models.py:class AgoraAPIError(APIError):
passerelle/contrib/agoraplus/models.py:class AgoraPlus(BaseResource):
passerelle/contrib/agoraplus/models.py:class AgoraPlusLink(models.Model):
passerelle/contrib/agoraplus/models.py:class AgoraPlusSas(models.Model):
```

Idem que pour `family`, j'ignore les données ?

```
passerelle/contrib/arcgis/models.py:class Arcgis(BaseResource):
passerelle/contrib/fake_family/models.py:class FakeFamily(BaseResource):
passerelle/contrib/greco/models.py:class Greco(BaseResource):
passerelle/contrib/iparapheur/models.py:class IParapheur(BaseResource):
passerelle/contrib/maarch/models.py:class Maarch(BaseResource):
passerelle/contrib/mdel/models.py:class MDEL(BaseResource):
passerelle/contrib/mdel/models.py:class Demand(models.Model):
```

Idem, pas de données ?

```
passerelle/contrib/meyzieu_newsletters/models.py:class MeyzieuNewsletters(BaseResource):
passerelle/contrib/nancypoll/models.py:class NancyPoll(BaseResource):
passerelle/contrib/seisin_by_email/models.py:class SeisinByEmailManagement(BaseResource):
passerelle/contrib/solis_apa/models.py:class SolisAPA(BaseResource):
passerelle/contrib/stub_invoices/models.py:class StubInvoicesConnector(BaseResource):
passerelle/contrib/teamnet_axel/models.py:class Management(BaseResource):
passerelle/contrib/teamnet_axel/models.py:class Link(models.Model):
```

Link ?

```
passerelle/contrib/tlmcom/models.py:class TlmCom(PBXMixin):
passerelle/datasources/models.py:class BaseDataSource(BaseResource):
passerelle/datasources/models.py:class CsvDataSource(BaseDataSource):
```

Je n'exporte pas je pense, c'est le modèle plus haut qui compte maintenant.

```
passerelle/pbx/models.py:class PBXMixin(BaseResource):
passerelle/queue/models.py:class BaseQueue(BaseResource):
passerelle/queue/models.py:class BaseOutputQueue(BaseQueue):
passerelle/queue/models.py:class JSONOutputQueue(BaseOutputQueue):
passerelle/queue/models.py:class MakoPrototypeOutputQueue(BaseOutputQueue):
passerelle/queue/models.py:class MakoOutputQueue(BaseOutputQueue):
```

Je n'exporte pas.

```
passerelle/repost/models.py:class BaseRepost(BaseResource):
```

Pas d'export (ça couvrira tous les Repost).

#6 - 16 janvier 2017 22:01 - Frédéric Péters

Je ne suis pas sûr de comprendre le formatage du commentaire, l'expression concerne le bloc ou seulement le dernier élément ? (je pencherais pour un mélange des deux et du coup c'est compliqué).

#7 - 16 janvier 2017 23:24 - Benjamin Dauvergne

Vérification que ApiUser.username est bien unique:

```
from hobo.multitenant.middleware import TenantMiddleware
from tenant_schemas.utils import tenant_context

from passerelle.base.models import ApiUser

for tenant in TenantMiddleware.get_tenants():
    with tenant_context(tenant):
        print tenant.domain_url, ': ',
        if ApiUser.objects.count() != len(set([apiuser.username for apiuser in ApiUser.objects.all()])):
            print 'NOK'
        else:
            print 'Ok'
```

En prod:

```
passerelle:/tmp# sudo -u passerelle passerelle-manage runscript test-apiuser-username.py
passerelle-publik.entrouvert.com : Ok
passerelle.nancy.fr : Ok
passerelle-meyzieu.entrouvert.org : Ok
passerelle.alfortville.fr : Ok
passerelle-calvados.entrouvert.org : Ok
passerelle.fontenay-sous-bois.fr : Ok
passerelle.chateauroux-metropole.fr : Ok
passerelle-fondettes.entrouvert.com : Ok
passerelle-vincennes.entrouvert.com : Ok
```

passerelle-comoe.entrouvert.com : Ok

En recette:

```
passerelle.test:/tmp# sudo -u passerelle passerelle-manage runscript test.py
passerelle-nancy.test.entrouvert.org : Ok
passerelle-gorges.test.entrouvert.org : Ok
passerelle-matrik.test.entrouvert.org : Ok
passerelle-clisson.test.entrouvert.org : Ok
passerelle-chateauroux.test.entrouvert.org : Ok
passerelle-meyzieu.test.entrouvert.org : Ok
passerelle-fsb.test.entrouvert.org : Ok
passerelle-calvados.test.entrouvert.org : Ok
passerelle2016.alfortville.fr : Ok
passerelle-csma.test.entrouvert.org : Ok
passerelle-fondettes.test.entrouvert.org : Ok
passerelle-vincennes.test.entrouvert.org : Ok
passerelle-orleans.test.entrouvert.org : Ok
passerelle-cch.test.entrouvert.org : Ok
passerelle-haute-goulaine.test.entrouvert.org : Ok
passerelle-rochefort.test.entrouvert.org : Ok
passerelle-blois.test.entrouvert.org : Ok
```

#8 - 16 janvier 2017 23:25 - Benjamin Dauvergne

Frédéric Péters a écrit :

Je ne suis pas sûr de comprendre le formatage du commentaire, l'expression concerne le bloc ou seulement le dernier élément ? (je pencherais pour un mélange des deux et du coup c'est compliqué).

Ça concerne la dernière application listée, l'avant dernier commentaire ne concerne que passerelle.queue par exemple.

#9 - 16 janvier 2017 23:55 - Benjamin Dauvergne

J'ai fait le même test sur AccessRight pour vérifier si (codename, apiuser_id, resource_type, resource_pk) forme une clé, et c'est presque vrai:

```
passerelle:/tmp# sudo -u passerelle passerelle-manage runscript test-apiuser-username.py
passerelle-publik.entrouvert.com : Ok
passerelle.nancy.fr : Ok
passerelle-meyzieu.entrouvert.org : Ok
passerelle.alfortville.fr : Ok
passerelle-calvados.entrouvert.org : Ok
passerelle.fontenay-sous-bois.fr : Ok
passerelle.chateauroux-metropole.fr : Ok
passerelle-fondettes.entrouvert.com : Ok
passerelle-vincennes.entrouvert.com : NOK
passerelle-comoe.entrouvert.com : Ok
```

```
passerelle.test:/tmp# sudo -u passerelle passerelle-manage runscript test2.py
passerelle-nancy.test.entrouvert.org : Ok
passerelle-gorges.test.entrouvert.org : Ok
passerelle-matrik.test.entrouvert.org : Ok
passerelle-clisson.test.entrouvert.org : Ok
passerelle-chateauroux.test.entrouvert.org : Ok
passerelle-meyzieu.test.entrouvert.org : Ok
passerelle-fsb.test.entrouvert.org : NOK
passerelle-calvados.test.entrouvert.org : Ok
passerelle2016.alfortville.fr : Ok
passerelle-csma.test.entrouvert.org : Ok
passerelle-fondettes.test.entrouvert.org : Ok
passerelle-vincennes.test.entrouvert.org : NOK
passerelle-orleans.test.entrouvert.org : Ok
passerelle-cch.test.entrouvert.org : Ok
passerelle-haute-goulaine.test.entrouvert.org : Ok
passerelle-rochefort.test.entrouvert.org : Ok
passerelle-blois.test.entrouvert.org : Ok
```

Le but étant d'ajouter un unique_together, toujours pour se prémunir de foirages dans un get_or_create.

#10 - 16 janvier 2017 23:59 - Benjamin Dauvergne

- Fichier 0001-create-unique-index-on-ApiUser.username-13887.patch ajouté

- Fichier 0002-implement-JSON-import-export-fixes-13887.patch ajouté

- Patch proposed changé de Non à Oui

Je pourrais essayer d'instantier plus de trucs dans les tests mais j'ai l'impression qu'à part CsvDataSource le reste nous intéresse assez peu.

#11 - 17 janvier 2017 10:29 - Benjamin Dauvergne

- Fichier 0001-create-unique-index-on-ApiUser.username-13887.patch ajouté

- Fichier 0002-create-unique-index-on-AccessRight.-13887.patch ajouté

- Fichier 0003-implement-JSON-import-export-fixes-13887.patch ajouté

Avec le unique_together sur AccessRight.*.

#12 - 17 janvier 2017 14:34 - Benjamin Dauvergne

Mon code empêche explicitement l'écrasement (si un slug existe déjà je ne fais rien) or dans le cas d'une migration de recette vers prod de connecteurs CSV on pourrait vouloir mettre à jour dans ceux-ci données et requêtes.

- Est-ce que j'ajoute à la méthode et la commande export-site un paramètre --resource-type permettant de n'exporter par exemple que les connecteurs CSV ?
- Est-ce que j'ajoute un flag --overwrite à import_site et import_json pour justement permettre une mise à jour (dans le cas du connecteur CSV on supprimerait toutes les requêtes avant de les recréer) ?

#13 - 17 janvier 2017 14:36 - Benjamin Dauvergne

Aussi je compte ajouter les propriétés suivantes au différents exports pour permettre facilement de rejeter un export.

```
'@type': 'passerelle-site'  
...  
'@type': 'passerelle-resource'  
...
```

Thomas me signale qu'exporter les APIUser et autres AccessRight n'a probablement aucun intérêt (car différent en prod et recette voir dans 99% inutile puisque la gestion des accès se fait autrement entre nos propres services).

#14 - 17 janvier 2017 14:46 - Serghei Mihai

Benjamin Dauvergne a écrit :

J'ai fait la liste des ressources ou autres modèles qui y sont liés:

[...]

Pour les connecteurs Teamnet et Agora je dirais que pas besoin d'exporter les données.

Le connecteur Teamnet ne garde que les liaisons nameid - (login, pwd) qui n'ont aucun intérêt à être copiées ailleurs.

Pareil pour le connecteur Agora, qui en plus contient des données dans le SaS sans intérêt à être copiées.

Pour le connecteur Family il peut être intéressant de copier les données sauf les associations.

#15 - 17 janvier 2017 14:48 - Thomas Noël

Serghei Mihai a écrit :

Pour le connecteur Family il peut être intéressant de copier les données

Mhhh.. dans quel cadre ? Pour moi non, il y a des loaders qui se chargent de cela. Non ?

#16 - 17 janvier 2017 14:51 - Serghei Mihai

Thomas Noël a écrit :

Mhhh.. dans quel cadre ? Pour moi non, il y a des loaders qui se chargent de cela. Non ?

Je vois un cas où on exporterait le connecteur et on voudrait tout de suite jouer avec les données, sans chercher à charger une archive. Mais c'est moi qui m'imagine trop de choses. Je ne m'oppose pas à ce qu'on n'exporte que les connecteurs et aucune donnée.

#17 - 17 janvier 2017 14:57 - Frédéric Péters

[...] que les connecteurs et aucune données.

Dans les cas précis cités. (parce que bien sûr pour le CSV, ou pour un connecteur montpellier/encombrants, on veut aussi les données).

#18 - 17 janvier 2017 15:53 - Benjamin Dauvergne

Frédéric Péters a écrit :

[...] que les connecteurs et aucune données.

Dans les cas précis cités. (parce que bien sûr pour le CSV, ou pour un connecteur montpellier/encombrants, on veut aussi les données).

Pour CSV je vois bien ce que sont les données mais pour encombrants (gdc?) je ne vois pas, il n'y qu'un modèle.

#19 - 17 janvier 2017 15:57 - Frédéric Péters

(j'imagine confusion encombrants vs gdc)

```
$ grep ^class passerelle_montpellier_encombrants/models.py
class EncombrantsManagement(BaseResource):
class Sector(models.Model):
class Commune(models.Model):
class Street(models.Model):
class CollectDay(models.Model):
```

#20 - 17 janvier 2017 16:43 - Benjamin Dauvergne

- Fichier 0001-implement-JSON-import-export-13887.patch ajouté

Je découvre ce projet :) Mais donc voilà un patch, je propose de ne pas exporter les jours de collecte, ça c'est vraiment de la donnée pour moi.

Au niveau des secteur, communes, rue, on fonctionne vraiment en écrasement cette fois.

#21 - 27 janvier 2017 00:10 - Benjamin Dauvergne

- Fichier 0003-implement-JSON-import-export-fixes-13887.patch ajouté

Patch principal mis à jour:

- j'ai renommé import-site/export-site en import_site/export_site (pour mieux coller au nom existant dans combo)
- par défaut je n'importe plus les utilisateurs ni les ACLs.

#22 - 27 janvier 2017 09:59 - Frédéric Péters

Reste dans le patch un # FIXME: Should we dump users ? I think it's dead.

Je pense pour ma part qu'il y a une certaine utilité, quand on a par exemple un connecteur CSV qu'on veut ouvrir au public, que ça survive l'export/import.

#23 - 27 janvier 2017 11:36 - Benjamin Dauvergne

Frédéric Péters a écrit :

Reste dans le patch un # FIXME: Should we dump users ? I think it's dead.

Commentaire invalide et plus à sa place du tout car je dump bien les utilisateurs mais dans passerelle.utils.export_site().

Je pense pour ma part qu'il y a une certaine utilité, quand on a par exemple un connecteur CSV qu'on veut ouvrir au public, que ça survive l'export/import.

J'ai conservé la fonctionnalité, export_site exporte toujours utilisateurs et ACLs, par contre par défaut import_site ne les importe pas (dans l'idée qu'import_site servira principalement à initialiser une plate-forme x depuis une plate-forme y, de test ou de dév). La commande import_site accepte un paramètre --import-users qui active cet import.

Il reste un point non traité, la question de l'écrasement, actuellement si un slug existe déjà je ne charge juste pas le dump correspondant. Est-ce que je permet un écrasement (paramètre --overwrite donné plus haut) ? Est-ce que ce doit être le mode de fonctionnement par défaut ? (je dirai que non).

Il faut voir ça en liaison avec la commande import-template apportée par hobo qui appellera import_site. Est-ce qu'on appelle import-template à chaque appel de déploiement ? Si non qu'est-ce qui fait qu'on ne l'appelle pas ? Si l'application est vide ? Comment mesurer ça génériquement, si

pas générique peut-être prévoir un flag --if-empty à la commande import_site.

Aussi sur le nom de la commande bien que combo ait commencé avec import_site je trouve que import-site est plus joli, je veux bien m'occuper de créer un alias coté combo.

#24 - 27 janvier 2017 11:44 - Frédéric Péters

Il reste un point non traité, la question de l'écrasement.

Côté combo, on écrase. Mais ça peut faire bizarre, parce que ça concerne évidemment uniquement les pages qui existeraient déjà, on ne supprime pas les autres. Manque sans doute une commande de remise à zéro.

Que ça soit par défaut ou non, je pense que c'est utile de pouvoir écraser; et que ça marche pareil d'une appli à l'autre (combo peut être modifié, bien sûr).

Est-ce qu'on appelle import-template à chaque appel de déploiement ?

Non, uniquement à la création initiale du tenant. Réflexion basée sur wcs, on peut avoir un squelette de site avec un échantillon de démarches, mais si j'en supprime une j'ai pas envie de la voir revenir.

Aussi sur le nom de la commande bien que combo ait commencé avec import_site je trouve que import-site est plus joli, je veux bien m'occuper de créer un alias coté combo.

La compat on peut l'ignorer ici, mais mais majoritairement on a des _ il me semble (create_tenant, create_hobo_tenant, hobo_notify, etc.). (et dans wcs un mix des deux).

#25 - 16 février 2017 15:53 - Jean-Baptiste Jaillet

Petit up sur le ticket, besoin pour le déploiement ozwillo du sictiam.

#26 - 03 mars 2017 16:49 - Benjamin Dauvergne

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

#27 - 03 mars 2017 16:50 - Benjamin Dauvergne

- Fichier 0001-import_site-add-clean-if-empty-and-override-options.patch ajouté

Voilà ce commit ajoute les options --clean qui vide passerelle avant d'importer, --if-empty qui ne charge que si le site est vide et --override qui écrase les connecteurs existant avec l'import (override ne nettoie pas les données attachés à un connecteur, pour le connecteur famille par exemple).

Je le rebaserai dans le patch 0003 ensuite.

#28 - 03 mars 2017 21:41 - Benjamin Dauvergne

Patch incomplet, voici le bon.

#29 - 03 mars 2017 21:42 - Benjamin Dauvergne

- Fichier 0001-import_site-add-clean-if-empty-and-override-options.patch ajouté

#30 - 05 mars 2017 10:18 - Frédéric Péters

Tu peux pousser ce qui est pertinent dans une branche ?

Notamment parce que :

Reste dans le patch un # FIXME: Should we dump users ? I think it's dead.

Commentaire invalide et plus à sa place du tout car je dump bien les utilisateurs mais dans passerelle.util.export_site().

Et qu'il n'y a pas de patch sans le FIXME attaché à ce ticket.

#31 - 06 mars 2017 10:56 - Benjamin Dauvergne

- Fichier 0003-implement-JSON-import-export-fixes-13887.patch ajouté

- Fichier 0004-import_site-add-clean-if-empty-and-override-options.patch ajouté

#32 - 06 mars 2017 10:59 - Benjamin Dauvergne

<http://git.entrouvert.org/passerelle/git/log/?h=wip/13887-export-import>

#33 - 06 mars 2017 11:53 - Frédéric Péters

Crash sur une FileField sans fichier (keystore du connecteur OrangeSMSSGateway dans mon cas).

```
File "/home/fred/src/eo/passerelle/passerelle/utils/__init__.py", line 189, in export_site
    resources.append(resource.export_json())
File "/home/fred/src/eo/passerelle/passerelle/base/models.py", line 218, in export_json
    'content': base64.b64encode(value.read()),
File "/home/fred/src/eo/venv/local/lib/python2.7/site-packages/django/core/files/utils.py", line 16, in <lambda>
    read = property(lambda self: self.file.read)
File "/home/fred/src/eo/venv/local/lib/python2.7/site-packages/django/db/models/fields/files.py", line 49, in
_n_get_file
    self._require_file()
File "/home/fred/src/eo/venv/local/lib/python2.7/site-packages/django/db/models/fields/files.py", line 46, in
_n_require_file
    raise ValueError("The '%s' attribute has no file associated with it." % self.field.name)
ValueError: The 'keystore' attribute has no file associated with it.
```

#34 - 06 mars 2017 12:01 - Frédéric Péters

J'ai ajouté un "continue" en cas de ValueError mais ce n'est peut-être pas suffisant, ou il y a un pendant à gérer à l'import,

```
File "/home/fred/src/eo/passerelle/passerelle/base/management/commands/import_site.py", line 29, in handle
    import_users=options['import_users'])
File "/home/fred/src/eo/passerelle/passerelle/utils/__init__.py", line 230, in import_site
    BaseResource.import_json(resource, overwrite=overwrite, import_users=import_users)
File "/home/fred/src/eo/passerelle/passerelle/base/models.py", line 246, in import_json
    instance = model.import_json_real(overwrite, instance, d)
File "/home/fred/src/eo/passerelle/passerelle/base/models.py", line 283, in import_json_real
    d[field.name]['name'],
KeyError: 'keystore'
```

#35 - 06 mars 2017 12:18 - Frédéric Péters

(à part ce soucis, c'est ok pour moi)

#36 - 06 mars 2017 13:10 - Benjamin Dauvergne

- Fichier 0003-implement-JSON-import-export-fixes-13887.patch ajouté

J'ai mergé les deux derniers patches, j'ai ajouté le traitement des FileField indéfini à l'export et à l'import et j'ai ajouté un test testant tout ça sur la création d'un modèle passerelle.apps.bdb.models.Bdb avec un keystore vide.

#37 - 06 mars 2017 13:15 - Benjamin Dauvergne

- Statut changé de Nouveau à Résolu (à déployer)

- % réalisé changé de 0 à 100

Appliqué par commit [6d3782a0ee005b0b838b0f48b30b2ce4bdd83ef4](https://git.entrouvert.org/passerelle/git/commit/6d3782a0ee005b0b838b0f48b30b2ce4bdd83ef4).

#38 - 06 mars 2017 13:17 - Frédéric Péters

Ok pour moi. (pour le debug avec pdb, je me rends compte que ça serait utile d'avoir un --output FILENAME, pour un autre ticket).

#39 - 04 août 2018 12:31 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-create-unique-index-on-ApiUser.username-13887.patch	1,75 ko	16 janvier 2017	Benjamin Dauvergne
0002-implement-JSON-import-export-fixes-13887.patch	16,9 ko	16 janvier 2017	Benjamin Dauvergne
0001-create-unique-index-on-ApiUser.username-13887.patch	1,75 ko	17 janvier 2017	Benjamin Dauvergne
0002-create-unique-index-on-AccessRight.-13887.patch	1,77 ko	17 janvier 2017	Benjamin Dauvergne
0003-implement-JSON-import-export-fixes-13887.patch	16,9 ko	17 janvier 2017	Benjamin Dauvergne
0001-implement-JSON-import-export-13887.patch	3,76 ko	17 janvier 2017	Benjamin Dauvergne

0003-implement-JSON-import-export-fixes-13887.patch	17,7 ko	26 janvier 2017	Benjamin Dauvergne
0001-import_site-add-clean-if-empty-and-overwrite-options.patch	6,72 ko	03 mars 2017	Benjamin Dauvergne
0001-import_site-add-clean-if-empty-and-overwrite-options.patch	9,46 ko	03 mars 2017	Benjamin Dauvergne
0003-implement-JSON-import-export-fixes-13887.patch	17,9 ko	06 mars 2017	Benjamin Dauvergne
0004-import_site-add-clean-if-empty-and-overwrite-options.patch	9,47 ko	06 mars 2017	Benjamin Dauvergne
0003-implement-JSON-import-export-fixes-13887.patch	21,2 ko	06 mars 2017	Benjamin Dauvergne