

django-mellon - Development #40833

Faciliter une logique de lookup adhoc par héritage

19 mars 2020 16:27 - Emmanuel Cazenave

Statut:	Fermé	Début:	19 mars 2020
Priorité:	Normal	Echéance:	
Assigné à:	Emmanuel Cazenave	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Dans la recherche par attribut on ne peut rechercher que sur le modèle qui vient via `auth.get_user_model()`.

En étant un tout petit plus permissif on pourrait franchir des relations.

Concrètement ça voudrait dire shooter :

```
try:
    User._meta.get_field(user_field)
except FieldDoesNotExist:
    logger.error('invalid LOOKUP_BY_ATTRIBUTES configuration %r, user field %s does not exist',
                line, user_field)
    continue
```

Ce qui permettrait de poser dans la conf des trucs comme ça :

```
MELLON_LOOKUP_BY_ATTRIBUTES = [
    {
        'saml_attribute': 'tabellio_pers_id',
        'user_field': 'docbowprofile__external_id',
    }
]
```

J'ai testé ça marche.

Révisions associées

Révision f16d316c - 31 mars 2020 15:10 - Emmanuel Cazenave

adapters: ease custom lookup through inheritance (#40833)

Historique

#2 - 19 mars 2020 16:43 - Benjamin Dauvergne

À voir si c'est vraiment plus intéressant de complexifier django-mellon ou de juste écrire un adapter pour docbow, qu'en penses-tu ?

Tu peux complètement remplacer `Adapter.lookup_user()` pour y faire ce que tu veux exactement sans te préoccuper d'éventuels changements dans django-mellon ou du fait que la configuration soit bien faite (si on est au pfwb c'est comme ci, si pw c'est comme ça).

#3 - 19 mars 2020 17:10 - Benjamin Dauvergne

À la rigueur, histoire de pas perdre le peu de debug qu'on a déjà sur ce point :

```
def check_field_path(base_klass, field_path):
    name = field_path.split('__', 1)[0]
    rest = field_path.split('__', 1)[1:]
    try:
        field = base_klass._meta.get_field(name)
    except FieldDoesNotExist:
        return False
    if rest:
        if not field.is_relation:
            return False
        return check_field_path(field.related_model, rest[0])
    ....
```

```
if not check_field_path(User, user_field):
    logger.error('invalid LOOKUP_BY_ATTRIBUTES configuration %r, user field %s does not exist',
                 line, user_field)
```

```
continue
```

#4 - 31 mars 2020 11:56 - Emmanuel Cazenave

Benjamin Dauvergne a écrit :

À voir si c'est vraiment plus intéressant de complexifier django-mellon ou de juste écrire un adapter pour docbow, qu'en penses-tu ?

En fait oui, c'est nettement plus simple, je préfère laisser tomber mon idée initiale.

Par contre pour ne pas avoir à tout re-écrire trop de choses dans un adapter spécifique, un petit hook du genre :

```
if lookup_by_attributes:
    user = self._lookup_by_attributes(idp, saml_attributes, lookup_by_attributes)
```

```
+ if user is None and hasattr(self, 'custom_lookup'):
+     user = self.cutom_lookup(idp, saml_attributes)
+
```

Ça t'irait ?

#5 - 31 mars 2020 12:30 - Benjamin Dauvergne

Hmmm idéalement il faudrait découper lookup_user qui en fait trop (ça fait lookup_by_nameid() or (lookup_by_attribute() and link) or create()).

Je réécrirait plutôt ce bout de code :

```
...
    user = None
    lookup_by_attributes = utils.get_setting(idp, 'LOOKUP_BY_ATTRIBUTES')
    if lookup_by_attributes:
        user = self._lookup_by_attributes(idp, saml_attributes, lookup_by_attributes)
...
```

comme cela :

```
def lookup_user_by_attributes(self, idp, saml_attributes):
    rules = utils.get_setting(idp, 'LOOKUP_BY_ATTRIBUTES'):
    if rules:
        ....
    return None
```

```
...
    def lookup_user(...):
    ....
        user = self.lookup_user_by_attributes(idp, saml_attributes)
    ...
```

T'auras plus qu'à surcharger ça et si ça renvoie None appliquer tes propres règles, wadoyouthink ?

#6 - 31 mars 2020 15:16 - Emmanuel Cazenave

- Fichier 0001-adapters-ease-custom-lookup-through-inheritance-4083.patch ajouté
- Sujet changé de Recherche par attribut : permette de franchir les relations à Faciliter une logique de look adhoc par héritage
- Statut changé de Nouveau à Solution proposée
- Assigné à mis à Emmanuel Cazenave
- Patch proposed changé de Non à Oui

Oui c'est plus souple comme ça, les classes filles peuvent choisir de donner la priorité à leur logique adhoc ou au lookup par attribut selon leur convenance.

#7 - 31 mars 2020 15:19 - Emmanuel Cazenave

- Sujet changé de Faciliter une logique de look adhoc par héritage à Faciliter une logique de lookup adhoc par héritage

#8 - 31 mars 2020 15:23 - Benjamin Dauvergne

- Statut changé de *Solution proposée* à *Solution validée*

#9 - 31 mars 2020 15:50 - Emmanuel Cazenave

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

```
commit f16d316cef5eb4bf7af2412fc11af6ba41e4037e
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date: Tue Mar 31 15:10:50 2020 +0200
```

```
adapters: ease custom lookup through inheritance (#40833)
```

#10 - 31 mars 2020 18:15 - Frédéric Péters

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

Fichiers

0001-adapters-ease-custom-lookup-through-inheritance-4083.patch	1,56 ko	31 mars 2020	Emmanuel Cazenave
---	---------	--------------	-------------------