

Authentic 2 - Development #67025

authentificateur saml : virer les JSONField

05 juillet 2022 14:21 - Valentin Deniaud

Statut:	Fermé	Début:	05 juillet 2022
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Il sont à remplacer par des objets en bonne et due forme (et ça va demander un peu de boulot).			

Révisions associées

Révision 0c2089c5 - 18 août 2022 18:34 - Valentin Deniaud

auth_saml: migrate JSON fields to models (#67025)

Révision bf63435a - 18 août 2022 18:34 - Valentin Deniaud

auth_saml: rename attributes using model (#67025)

Révision b1f6ddcd - 18 août 2022 18:34 - Valentin Deniaud

auth_saml: set attributes using model (#67025)

Révision 1ea8c999 - 18 août 2022 18:34 - Valentin Deniaud

auth_saml: add roles using model and remove useless code (#67025)

Révision e776500f - 18 août 2022 18:34 - Valentin Deniaud

auth_saml: lookup by attributes using model (#67025)

Révision b486ca00 - 18 août 2022 18:34 - Valentin Deniaud

auth_saml: remove JSON fields from model (#67025)

Révision 18ea92a7 - 18 août 2022 18:34 - Valentin Deniaud

manager: factorize role with OU display code (#67025)

Révision 742e955d - 18 août 2022 18:34 - Valentin Deniaud

auth_saml: add views to configure related objects (#67025)

Historique

#1 - 27 juillet 2022 11:14 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

#2 - 27 juillet 2022 11:42 - Benjamin Dauvergne

Je ne suis pas entièrement convaincu, dans quel cas on voudra requêter ces objets ? Si c'est juste parce qu'un modèle te donne la génération automatique des formulaires ça ne me semble pas super utile.

#3 - 27 juillet 2022 12:15 - Valentin Deniaud

Benjamin Dauvergne a écrit :

Je ne suis pas entièrement convaincu, dans quel cas on voudra requêter ces objets ? Si c'est juste parce qu'un modèle te donne la génération automatique des formulaires ça ne me semble pas super utile.

C'est vraiment nul comme interface d'avoir à taper du JSON, se rappeler de la syntaxe bizarre, ne pas faire de faute de frappe.

On y gagnera aussi en terme d'affichage des infos, de validation, de facilité d'évolutions et de clarification du code qui parse ça.

#4 - 27 juillet 2022 12:55 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Benjamin Dauvergne a écrit :

Je ne suis pas entièrement convaincu, dans quel cas on voudra requêter ces objets ? Si c'est juste parce qu'un modèle te donne la génération automatique des formulaires ça ne me semble pas super utile.

C'est vraiment nul comme interface d'avoir à taper du JSON, se rappeler de la syntaxe bizarre, ne pas faire de faute de frappe.

C'est là que je ne comprends pas, c'est quoi le rapport entre taper du JSON et stocker en JSON ? Tu peux faire des formulaires qui produisent du JSON c'est pas un souci.

PS: des trucs simples comme ça (pseudocode, je copie vaguement du code de la doc des FormSet de Django)

```
class LookupByAttributes(Form):
    x = forms.CharField()
    y = forms.CharField()

def edit_lookup_by_attributes(request, pk):
    authenticator = get_object_or_404(SAMLAuthenticator, pk=pk)
    LookupByAttributesFormSet = forms.formset_factory(SAMLLookupByAttributes)
    formset = LookupByAttributesFormSet(data=request.GET)
    if formset.is_valid():
        authenticator.lookup_by_attributes = list(formset.cleaned_data)
        authenticator.save()
        return redirect('authenticator-details', kwargs={'pk': pk})
    return render(request, 'edit.html', context={'formset': formset})
```

#5 - 27 juillet 2022 13:47 - Valentin Deniaud

Benjamin Dauvergne a écrit :

C'est là que je ne comprends pas, c'est quoi le rapport entre taper du JSON et stocker en JSON ? Tu peux faire des formulaires qui produisent du JSON c'est pas un souci.

OK je n'avais pas compris que ta remarque portait sur avoir des objets et pas sur avoir des formulaires.

Stocker du JSON c'est quand même moins dans les clous que de stocker un objet.

PS: des trucs simples comme ça (pseudocode, je copie vaguement du code de la doc des FormSet de Django) [...]

Oui et tu prends LOOKUP_BY_ATTRIBUTES qui est simple, par contre A2_ATTRIBUTE_MAPPING c'est un bordel qui contient à la fois le mapping et aussi des actions pour renommer des attributs et ajouter des rôles. Je voudrais une séparation claire dans l'interface entre les 3, or garder du JSON ça voudrait dire bricoler un gros formulaire qui permette de configurer tout, on ne pourrait pas avoir 3 formulaires séparés notamment parce que l'ordre des actions dans le JSON importe.

Bref si l'argument c'est l'inutilité de passer du temps à migrer ces JSONField dans des objets, je suis convaincu que je vais passer moins de temps à le faire que ce que tu me proposes.

#6 - 27 juillet 2022 16:27 - Benjamin Dauvergne

Valentin Deniaud a écrit :

PS: des trucs simples comme ça (pseudocode, je copie vaguement du code de la doc des FormSet de Django) [...]

Oui et tu prends LOOKUP_BY_ATTRIBUTES qui est simple, par contre A2_ATTRIBUTE_MAPPING c'est un bordel qui contient à la fois le mapping et aussi des actions pour renommer des attributs et ajouter des rôles. Je voudrais une séparation claire dans l'interface entre les 3, or garder du JSON ça voudrait dire bricoler un gros formulaire qui permette de configurer tout, on ne pourrait pas avoir 3 formulaires séparés notamment parce que l'ordre des actions dans le JSON importe.

La façon dont c'est stocké n'a aucune influence sur la façon dont c'est présenté dans l'IHM ni dans la façon dont c'est transmis à django-mellon, coté w.c.s. on a tout un workflow dans un seul objet pickle qu'on pourrait apparenter à un document JSON et ça n'empêche pas d'éditer ça comme on veut.

Si tu ne veux pas utiliser des FormSet mais des formulaires indépendants pour éditer et ajouter un nouvel "ATTRIBUTE_MAPPING" rien ne t'en empêche, si tu trouves plus simples d'avoir un identifiant par ligne pour l'édition tu peux l'ajouter. Tu peux vraiment écrire ça de la même manière que tu l'aurais fait avec des modèles.

Idem si tu veux séparer les actions en plusieurs groupes avec des formulaires différents tu peux aussi.

#7 - 27 juillet 2022 16:59 - Valentin Deniaud

Ce que je voulais dire c'est que

```
A2_ATTRIBUTE_MAPPING = [{'action': 'rename', ...}, {'action': 'set-attribute', ...}]
```

si je veux avoir un formulaire spécialisé pour ajouter/éditer une action renommage, ok je peux lire/écrire dans le JSONField mais il va falloir faire attention à l'ordre vu qu'un set-attribute dépend du rename, ça reviendra à mettre toujours les renames en début de liste et c'est de la bidouille.

Mon argument c'est vraiment qu'on gagnera à structurer les choses sans stockage JSON, le code sera plus explicite, plus lisible, et le monde plus beau.

#8 - 27 juillet 2022 17:27 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Ce que je voulais dire c'est que
[...]

Tu réécis ça en @A2_ATTRIBUTE_MAPPING = [{"uuid": "abcde...", }, {"uuid": "12345", ...}] et magie tu peux adresser chacun d'entre eux sans te soucier de l'ordre qui sera conservé via

```
for row in model.a2_attribute_mapping:
    if row['uuid'] == uuid:
        row.clear()
        row.update(new_row)
model.save()
```

C'est exactement ce qui est fait pour les champs des formulaires ou le statuts dans w.c.s. et personne n'y trouve d'inconvénient (modulo l'allocation des id qui a été longtemps foireuse, partir sur de l'uuid c'est être certain de n'avoir jamais de souci).

si je veux avoir un formulaire spécialisé pour ajouter/éditer une action renommage, ok je peux lire/écrire dans le JSONField mais il va falloir faire attention à l'ordre vu qu'un set-attribute dépend du rename, ça reviendra à mettre toujours les renames en début de liste

Je ne comprends pas pourquoi tu penses cela, le formulaire affiché dépend de la valeur de row['action'] c'est à peu près tout, pour réordonner tu postes la listes des UUIDs dans l'ordre, idem pour supprimer, y a vraiment aucun souci.

Mon argument c'est vraiment qu'on gagnera à structurer les choses sans stockage JSON, le code sera plus explicite, plus lisible, et le monde plus beau.

C'est déjà structuré, re-structurer autrement c'est un boulot inutile, faut juste présenter les choses.

#10 - 16 août 2022 14:12 - Valentin Deniaud

- Fichier 0004-auth_saml-add-roles-using-model-and-remove-useless-c.patch ajouté
- Fichier 0005-auth_saml-lookup-by-attributes-using-model-67025.patch ajouté
- Fichier 0007-auth_saml-add-views-to-configure-related-objects-670.patch ajouté
- Fichier 0001-auth_saml-migrate-JSON-fields-to-models-67025.patch ajouté
- Fichier 0006-auth_saml-remove-JSON-fields-from-model-67025.patch ajouté
- Fichier 0003-auth_saml-set-attributes-using-model-67025.patch ajouté
- Fichier 0002-auth_saml-rename-attributes-using-model-67025.patch ajouté
- Tracker changé de Support à Development
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

0001 : migration des JSON fields dans des modèles (get_role/get_ou sont copiées/collées depuis adapters.py)

0002 -> 0005 : utilisation des nouveaux champs

0006 : suppression des champs inutilisés

0007 : ajout des vues (les onglets font leur entrée dans authentic !)

#11 - 16 août 2022 15:00 - Thomas Noël

Dans le code de migration :

```
attribute = get_key(obj, 'attribute', 32)
saml_attribute = get_key(obj, 'saml_attribute', 128)
```

qui va faire que la config va être complètement explosée si on a des configurations en JSON qui dépassent ces limites. Je propose :

- de pousser à 1024 partout. Même pas peur, il y aura peu de lignes. Ainsi, l'assurance à 99,9999% de n'avoir jamais de coupure. Si c'est jugé abusif, 512, allez.
- si possible d'émettre un warning/trace dans la migration get_key quand ça dépasse (mais franchement ça n'arrivera pas, donc si on sait pas faire de warning)

(Désolé, j'arrête ici, courte relecture)

#12 - 16 août 2022 15:21 - Thomas Noël

Par rapport à AddRoleAction :

- je considère que le mandatory sur AddRoleAction est inutile (et inutilisable). Si la condition plante, ne donnons pas le rôle. Retirer donc ce flag.
- Sur condition, j'ajouterais un hint, comme pense-bête, pour rappeler que si la condition n'est pas respectée alors le rôle est retiré.
- et en vérité je serais pour retirer complètement "condition", qui n'a jamais été utilisé... et ne le sera jamais.

#13 - 16 août 2022 15:57 - Thomas Noël

Désolé, relecture hachée...

Toujours dans 0001

- Sur le SAMLAttributeLookup ça serait bien d'avoir un choix au niveau du user_field (ou une ForeignKey, mais je doute qu'on puisse).
- Juste par esthétique, mettre le modèle RenameAttributeAction au dessus de SAMLAttributeLookup pour se rappeler qu'on fait d'abord le rename avec le lookup.
- À ce sujet, pourquoi avoir nommé SAMLAttributeLookup et pas SAMLAttributeMapping ? (vraiment sans importance, juste de la curiosité perso).

Plus loin, le « wrap_mapping_errors » donne du code plus court mais... c'est moins facile à lire pour quelqu'un comme moi. Et j'ai l'impression qu'on va perdre dans cette bataille les erreurs explicites « missing saml_attribute key » et « missing attribute key » et c'est très dommage. Comme je propose plus haut de flinguer le mandatory sur AddRoleAction, je pense que tu pourrais revoir l'affaire sans ce contextmanager.

Sur le formulaire AddRoleAction, en cas de multicollectivité, il faudrait que la liste des rôles ajoute l'OU de chacun entre parenthèse (ou autre).

C'est très très mal de ne pas nommer les migrations :)

SAMLAuthenticatorEventsMixin gagnerait dans mon estime à être renommé SAMLAuthenticatorEventsBase ou SAMLAuthenticatorEvents

Voilà, c'est tout.

Ensuite, je pense qu'après avoir retiré la gestion de condition du AddRoleAction, on peut aussi supprimer l'action de renommage des attributs SAML, qui ne servira plus. Mais c'est plutôt pour un autre ticket "simplification"...

#14 - 17 août 2022 11:43 - Valentin Deniaud

- Fichier 0008-auth_saml-add-views-to-configure-related-objects-670.patch ajouté
- Fichier 0004-auth_saml-add-roles-using-model-and-remove-useless-c.patch ajouté
- Fichier 0005-auth_saml-lookup-by-attributes-using-model-67025.patch ajouté
- Fichier 0001-auth_saml-migrate-JSON-fields-to-models-67025.patch ajouté
- Fichier 0006-auth_saml-remove-JSON-fields-from-model-67025.patch ajouté
- Fichier 0003-auth_saml-set-attributes-using-model-67025.patch ajouté
- Fichier 0002-auth_saml-rename-attributes-using-model-67025.patch ajouté
- Fichier 0007-manager-factorize-role-with-OU-display-code-67025.patch ajouté

Thomas Noël a écrit :

Je propose :

- de pousser à 1024 partout.

Yep, j'ai fait ça.

- si possible d'émettre un warning/trace dans la migration get_key quand ça dépasse (mais franchement ça n'arrivera pas, donc si on sait pas faire de warning)

Pas certain que ça marche et compliqué à tester, je passe.

Par rapport à AddRoleAction :

- je considère que le mandatory sur AddRoleAction est inutile (et inutilisable). Si la condition plante, ne donnons pas le rôle. Retirer donc ce flag.
- Sur condition, j'ajouterais un hint, comme pense-bête, pour rappeler que si la condition n'est pas respectée alors le rôle est retiré.
- et en vérité je serais pour retirer complètement "condition", qui n'a jamais été utilisé... et ne le sera jamais.

J'ai supprimé les mentions de mandatory et condition ainsi que les comportements associés, par contre je les garde dans le modèle et la migration, dans l'idée que si quelqu'un s'en servait et s'en plaint, il y aura juste à remettre le code sans migration relou à écrire.

- Sur le SAMLAttributeLookup ça serait bien d'avoir un choices au niveau du user_field (ou une ForeignKey, mais je doute qu'on puisse).

C'est plus compliqué qu'il n'y paraît parce que les attributs viennent du modèle Attribute (sauf ceux natifs à Django ?), je verrais bien ça pour un autre ticket.

- Juste par esthétique, mettre le modèle RenameAttributeAction au dessus de SAMLAttributeLookup pour se rappeler qu'on fait d'abord le rename avec le lookup.

Fait.

- À ce sujet, pourquoi avoir nommé SAMLAttributeLookup et pas SAMLAttributeMapping ? (vraiment sans importance, juste de la curiosité perso).

On a deux modèles similaires, l'un pour rechercher par attribut l'autre pour affecter des valeurs aux attributs. SAMLAttributeMapping je trouve qu'on ne sait pas auquel des deux il correspond, SAMLAttributeLookup ça me semble plus évident (mais la raison de base c'est qu'on migre dans ce modèle la valeur du setting LOOKUP_BY_ATTRIBUTES, j'ai juste repris cette terminologie).

Plus loin, le « wrap_mapping_errors » donne du code plus court mais... c'est moins facile à lire pour quelqu'un comme moi.

Retiré.

Et j'ai l'impression qu'on va perdre dans cette bataille les erreurs explicites « missing saml_attribute key » et « missing attribute key » et c'est très dommage.

Non c'est normal, ce sont des erreurs qui traduisent une configuration invalide qui ne peut plus exister avec les modèles.

Sur le formulaire AddRoleAction, en cas de multicollectivité, il faudrait que la liste des rôles ajoute l'OU de chacun entre parenthèse (ou autre).

Ouep, on a déjà ça ailleurs dans le manager, je reprends le code.

C'est très très mal de ne pas nommer les migrations :)

Renommé.

SAMLAuthenticatorEventsMixin gagnerait dans mon estime à être renommé SAMLAuthenticatorEventsBase ou SAMLAuthenticatorEvents

Fait.

Ensuite, je pense qu'après avoir retiré la gestion de condition du AddRoleAction, on peut aussi supprimer l'action de renommage des attributs SAML, qui ne servira plus. Mais c'est plutôt pour un autre ticket "simplification"...

Yep, ça fera un deuxième ticket à ouvrir.

Et merci pour la relecture !

#15 - 18 août 2022 18:27 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Une seule remarque, au niveau de

```
class AddRoleAction(SAMLRelatedObjectBase):
    role = models.ForeignKey(Role, verbose_name=_('Role'), on_delete=models.CASCADE)
    condition = models.CharField(editable=False, max_length=256, blank=True)
```

```
mandatory = models.BooleanField(editable=False, default=False)
```

je poserais des `verbose_name=_('... [obsolete/unused]')` sur condition et mandatory pour rappeler dans le code et dans une éventuelle interface admin que non, ces choses sont là juste pour la gloire. Ca pourrait être juste un commentaire, aussi. Ou tu laisses ainsi et ce ticket rappellera qu'on ne gère plus ni l'un ni l'autre pour l'attribution des rôles.

Je valide néanmoins : tu fais comme tu le sens.

#16 - 18 août 2022 18:34 - Valentin Deniaud

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

```
commit 53ac88099c2486f630f4f47c9e7e54a180d6757b
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Tue Aug 16 17:11:14 2022 +0200
```

auth_saml: add views to configure related objects (#67025)

```
commit c9908d67dbbe4f1ed162c71ffd1f2f7fe4da47db
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Wed Aug 17 10:09:12 2022 +0200
```

manager: factorize role with OU display code (#67025)

```
commit 00c4614fca71cbd9169094203520cd8c8e4b21df
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Jul 28 18:18:31 2022 +0200
```

auth_saml: remove JSON fields from model (#67025)

```
commit 41ee78d6285ab9af2ac4570fad66af87add4cd51
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Jul 28 15:52:33 2022 +0200
```

auth_saml: lookup by attributes using model (#67025)

```
commit 0280a6114bc4a6ecd0ace69b60b915ddf95595d8
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Jul 28 15:50:20 2022 +0200
```

auth_saml: add roles using model and remove useless code (#67025)

```
commit 62c11d362438542ccdedfeb6216999ff7e0774d8
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Jul 28 15:32:34 2022 +0200
```

auth_saml: set attributes using model (#67025)

```
commit 52d189cb5de7c22066ef4760ef1a73818c00bb0b
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Jul 28 15:18:19 2022 +0200
```

auth_saml: rename attributes using model (#67025)

```
commit ec3398d3a16445709bc7be39c3d2d9e440f6cdaf
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Wed Jul 27 16:21:40 2022 +0200
```

auth_saml: migrate JSON fields to models (#67025)

#17 - 18 août 2022 20:14 - Transition automatique

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

#18 - 23 octobre 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0004-auth_saml-add-roles-using-model-and-remove-useless-c.patch	11,3 ko	16 août 2022	Valentin Deniaud
0005-auth_saml-lookup-by-attributes-using-model-67025.patch	2,36 ko	16 août 2022	Valentin Deniaud
0007-auth_saml-add-views-to-configure-related-objects-670.patch	29,1 ko	16 août 2022	Valentin Deniaud

0001-auth_saml-migrate-JSON-fields-to-models-67025.patch	18 ko	16 août 2022	Valentin Deniaud
0006-auth_saml-remove-JSON-fields-from-model-67025.patch	2,66 ko	16 août 2022	Valentin Deniaud
0003-auth_saml-set-attributes-using-model-67025.patch	6,75 ko	16 août 2022	Valentin Deniaud
0002-auth_saml-rename-attributes-using-model-67025.patch	5,41 ko	16 août 2022	Valentin Deniaud
0008-auth_saml-add-views-to-configure-related-objects-670.patch	27,1 ko	17 août 2022	Valentin Deniaud
0004-auth_saml-add-roles-using-model-and-remove-useless-c.patch	12,1 ko	17 août 2022	Valentin Deniaud
0005-auth_saml-lookup-by-attributes-using-model-67025.patch	2,35 ko	17 août 2022	Valentin Deniaud
0001-auth_saml-migrate-JSON-fields-to-models-67025.patch	17,8 ko	17 août 2022	Valentin Deniaud
0006-auth_saml-remove-JSON-fields-from-model-67025.patch	2,65 ko	17 août 2022	Valentin Deniaud
0003-auth_saml-set-attributes-using-model-67025.patch	6,31 ko	17 août 2022	Valentin Deniaud
0002-auth_saml-rename-attributes-using-model-67025.patch	5,41 ko	17 août 2022	Valentin Deniaud
0007-manager-factorize-role-with-OU-display-code-67025.patch	2,45 ko	17 août 2022	Valentin Deniaud