Authentic 2 - Development #36377

API pour définir les membres (directs) d'un rôle

24 septembre 2019 07:44 - Frédéric Péters

Statut:	Fermé	**Début:**	24 septembre 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Paul Marillonnet	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

On a une API permettant d'ajouter(/retirer) un utilisateur à un rôle, roles/(?P<role_uuid>[\w+]*)/members/(?P<member_uuid>[^/]+)/.

On pourrait avoir une API roles/(?P<role_uuid>[\w+]*)/members/, où un POST d'une liste de <member_uuid> ajouterait tout ceux-ci, un DELETE retirerait tout ceux-ci, mais surtout, un PUT viderait/définirait la liste des membres ?

Révisions associées

Révision 1cedef29 - 04 octobre 2019 18:49 - Paul Marillonnet

api: role members direct definition (#36377)

Historique

#1 - 24 septembre 2019 12:01 - Benjamin Dauvergne

Ça m'irait qu'on rapproche l'API du fonctionnement de JSONAPI

```
# ajouter l'utilisateur 1234
POST /api/roles/xyz/relationships/members/

{
   "data": [
     {"uuid": "1234"}
   ]
}
# écraser la liste des membres avec les utilisateurs pointés
PATCH /api/roles/xyz/relationships/members/

{
   "data": [
     {"uuid": "1234"}
   ]
}

# retirer l'utilisateur 1234
DELETE /api/roles/xyz/relationships/members/

{
   "data": [
     {"uuid": "1234"}
   ]
}
```

L'idée étant que /api/roles/xyz/members/ fonctionne par contre comme /api/users/ un jour (et faut prendre en compte le cas {"sub": "1234"} pour les services OIDCs.

L'ancienne API serait déprécié progressivement.

#2 - 24 septembre 2019 12:30 - Benjamin Dauvergne

- Assigné à mis à Nicolas Roche

#3 - 24 septembre 2019 15:15 - Paul Marillonnet

- Assigné à changé de Nicolas Roche à Paul Marillonnet

Je vais prendre celui-là.

#4 - 24 septembre 2019 15:18 - Paul Marillonnet

Que fait-on lorsqu'un des uuid dans le payload n'existe pas pour a2 ?
J'imagine que le comportement actuel de RoleMembershipsAPI, qui consiste à renvoyer un HTTP 404, n'est plus valable.
Au contraire, on pourrait renvoyer la liste des UUID d'usagers qui ont été affectés par l'appel.

#5 - 24 septembre 2019 15:28 - Frédéric Péters

Retourner le nouveau contenu, genre ce que pourrait retourner GET /api/roles/xyz/relationships/members/ qui n'était pas mentionné, ça me semble raisonnable.

#6 - 24 septembre 2019 16:40 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Que fait-on lorsqu'un des uuid dans le payload n'existe pas pour a2 ?

Je serai pour renvoyer une erreur 400 par défaut, si c'est gênant on inventer un paramètre pour débrayer ; un truc permissif n'obligera pas les gens à valider le retour.

> J'imagine que le comportement actuel de RoleMembershipsAPI, qui consiste à renvoyer un HTTP 404, n'est plus valable.

Non et c'était moche.

> Au contraire, on pourrait renvoyer la liste des UUID d'usagers qui ont été affectés par l'appel.

Si tout se passe bien normalement ce serait 204 No content.

#7 - 26 septembre 2019 16:36 - Paul Marillonnet

- Fichier 0001-api-direct-role-members-definition-36377.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Une première version, qui reprend ce qui est fait pour l'API /roles/<role_uuid>/members/<member_uuid>/, en ajoutant du code dans la classe associée.
Elle reprend en particulier l'idée de renvoyer un 20{0,1} quand ça se passe bien, avec une courte phrase de confirmation de l'opération effectuée.

#8 - 26 septembre 2019 16:53 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Une première version, qui reprend ce qui est fait pour l'API /roles/<role_uuid>/members/<member_uuid>/, en ajoutant du code dans la classe associée.
> Elle reprend en particulier l'idée de renvoyer un 20{0,1} quand ça se passe bien, avec une courte phrase de confirmation de l'opération effectuée.

Je préférerai une classe séparée pour pouvoir déprécier l'autre à un moment et pour l'URL /api/roles/<role_uuid>/relationships/members/.

#9 - 26 septembre 2019 17:02 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Benjamin Dauvergne a écrit :

> Je préférerai une classe séparée pour pouvoir déprécier l'autre à un moment et pour l'URL /api/roles/<role_uuid>/relationships/members/.

Oui, ce sera plus propre comme ça.

#10 - 26 septembre 2019 17:36 - Paul Marillonnet

- Fichier 0001-api-direct-role-members-definition-36377.patch ajouté

- Statut changé de En cours à Solution proposée

Voilà, c'est mieux comme ça, en effet.

#11 - 27 septembre 2019 11:25 - Paul Marillonnet

- Fichier 0001-api-do-not-give-uuid-validity-info-to-unauthorized-r.patch ajouté

- Fichier 0002-api-explicit-role-membership-addition-removal-testin.patch ajouté

- Fichier 0003-api-role-members-direct-definition-36377.patch ajouté

En séparant un peu mieux les choses.

#12 - 02 octobre 2019 12:03 - Frédéric Péters

Je ne sais pas ce qu'authentic ou DRF font côté transactions, et du coup je suggérerais des explicites .atomic(). (mais me répondre que c'est bon c'est géré en amont ça me va très bien).

#13 - 02 octobre 2019 15:31 - Lauréline Guérin

```
    def patch(self, request, *args, **kwargs):
        self.role.members.all().delete()
        for member in self.members:
            self.role.members.add(member)
```

et pourquoi pas juste

```
    def patch(self, request, *args, **kwargs):
        self.role.members.set(members)
```

?

#14 - 02 octobre 2019 16:15 - Paul Marillonnet

- Fichier 0001-api-role-members-direct-definition-36377.patch ajouté

Lauréline Guerin a écrit :

> et pourquoi pas juste

Bien vu, merci.

De façon similaire, usage des fonctions d'ajout et de retrait du ManyRelatedManager de l'ORM de django. Les questions d'atomicité dans la vue de l'API ne se posent donc plus.

#15 - 02 octobre 2019 16:33 - Paul Marillonnet

(tests en django 1.8 qui cassent mais qui cependant n'ont plus lieu d'être, je viens de pousser une branche rebasée sur un master à jour)

#16 - 03 octobre 2019 10:40 - Benjamin Dauvergne

C'est PATCH -> add et PUT/POST -> set.

#17 - 03 octobre 2019 13:53 - Paul Marillonnet

- Fichier 0001-api-role-members-direct-definition-36377.patch ajouté

Discussion sur le salon tech à ce sujet, il s'agit d'une collection donc c'est bien POST -> add et PATCH/PUT -> set.

Nouveau patch avec :

- ajout de /relationships/ dans le motif d'uri
- test de l'absence de doublon
- retrait de la fixture paramétrée simple_user qui provoquait inutilement une explosion combinatoire (les tests écrits dans ce patch étaient déclinés en 96 appels différents...)

#18 - 04 octobre 2019 02:14 - Benjamin Dauvergne

Je ne suis pas d'accord avec le 404 quand on recherche les UUIDs, il faut renvoyer une 400 dans tous les cas, une ValidationError expliquant l'uuid qui foire.

#19 - 04 octobre 2019 09:32 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Benjamin Dauvergne a écrit :

> Je ne suis pas d'accord avec le 404 quand on recherche les UUIDs, il faut renvoyer une 400 dans tous les cas, une ValidationError expliquant l'uuid qui foire.

Ok ça me va. D'ailleurs c'est hors specs JSONAPI[1].

[1] https://jsonapi.org/format/#crud-updating-relationship-responses :

```
 200 OK
[…]

403 Forbidden
[…]

A server MAY respond with other HTTP status codes.
```

#20 - 04 octobre 2019 10:15 - Paul Marillonnet

- Fichier 0001-api-role-members-direct-definition-36377.patch ajouté

- Statut changé de En cours à Solution proposée

#21 - 04 octobre 2019 10:21 - Paul Marillonnet

- Fichier 0001-api-role-members-direct-definition-36377.patch ajouté

La même chose en prêtant attention à la longueur maximale des lignes dans le code de la vue.

#22 - 04 octobre 2019 10:50 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Benjamin Dauvergne a écrit :
>
> > Je ne suis pas d'accord avec le 404 quand on recherche les UUIDs, il faut renvoyer une 400 dans tous les cas, une ValidationError expliquant l'uuid qui foire.

Sans aller jusqu'à relire le document JSONAPI (je veux pas en faire une bible, mais on a pas de référence sur comment structurer nos APIs pour l'instant à part renvoyer {'err':...} et donc on fait un peut chaque fois des choses différentes... bref) 404 c'est que la ressource n'existe pas, la ressource existe, par contre la requête est bien erronée, donc 400.

#23 - 04 octobre 2019 10:52 - Benjamin Dauvergne

Manque encore un check que request.data est bien un dict (tu te serais moins emmerdé avec un Serializer).

#24 - 04 octobre 2019 11:40 - Paul Marillonnet

- Fichier 0001-api-role-members-direct-definition-36377.patch ajouté

Benjamin Dauvergne a écrit :

> Manque encore un check que request.data est bien un dict (tu te serais moins emmerdé avec un Serializer).

Voilà.

#25 - 04 octobre 2019 18:41 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

#26 - 04 octobre 2019 19:02 - Paul Marillonnet

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

```
commit 1cedef29c9d4973061b7731626e390c985770bb8
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Fri Sep 27 11:24:13 2019 +0200

    api: role members direct definition (#36377)
```

#27 - 06 octobre 2019 10:15 - Frédéric Péters

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

Fichiers

0001-api-direct-role-members-definition-36377.patch	12,2 ko	26 septembre 2019	Paul Marillonnet
0001-api-direct-role-members-definition-36377.patch	12,4 ko	26 septembre 2019	Paul Marillonnet
0001-api-do-not-give-uuid-validity-info-to-unauthorized-r.patch	2,23 ko	27 septembre 2019	Paul Marillonnet
0002-api-explicit-role-membership-addition-removal-testin.patch	1,21 ko	27 septembre 2019	Paul Marillonnet
0003-api-role-members-direct-definition-36377.patch	10,4 ko	27 septembre 2019	Paul Marillonnet
0001-api-role-members-direct-definition-36377.patch	10,3 ko	02 octobre 2019	Paul Marillonnet
0001-api-role-members-direct-definition-36377.patch	9,91 ko	03 octobre 2019	Paul Marillonnet
0001-api-role-members-direct-definition-36377.patch	10,3 ko	04 octobre 2019	Paul Marillonnet
0001-api-role-members-direct-definition-36377.patch	10,6 ko	04 octobre 2019	Paul Marillonnet
0001-api-role-members-direct-definition-36377.patch	11,3 ko	04 octobre 2019	Paul Marillonnet