

## Passerelle - Development #5124

### autorisations d'accès - ui

10 juillet 2014 11:41 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	10 juillet 2014
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Frédéric Péters	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	
<b>Patch proposed:</b>	Oui		

**Description**

Dans un schéma simple, il y aurait d'une part manage/access/ pour créer/lister/éditer/supprimer les ApiUser, et dans les vues des ressources une section "Sécurité" qui listerait les permissions et permettrait d'y attacher les ApiUser.

À réfléchir à partir de cette UI , j'ai l'impression qu'on est passé à côté d'un truc, qu'on n'arrive pas à modéliser la situation où l'apiuser X a la permission A sur un service de type M, puis la permission B sur un autre service de type M, et qu'on aurait mieux fait d'avoir plusieurs manytomany au niveau des ressources, selon les différentes permissions.

```
class Gdc(...):
    can_post_users = models.ManyToManyField(ApiUser, blank=True)
    can_view_users = models.ManyToManyField(ApiUser, blank=True)
```

L'objet Permission de Django, il n'est pas adapté, en fait.

### Historique

#### #1 - 10 juillet 2014 16:31 - Thomas Noël

Effectivement. Comme quoi, même à 4, on voit pas toujours tout... Mais je trouverai un peu (beaucoup) dommage d'avoir une table par permission -- ce qui sera le cas si on déclare autant de ManyToMany que de permission tel que proposé.

(bon, pour limiter à une table de permission par type d'objet, on pourrait avoir un seul ManyToMany dans la classe GDC, utilisant un through=Permission, et dans le modèle Permission on aurait (gdc, apiuser, permission)... mais c'est encore un peu moche).

Idéalement, je pensais à une seule table générale à toute la passerelle : {object\_type, object\_pk, apiuser\_id, permission}, par exemple la ligne {'gdc', '3', '4', 'can\_post'} dirait que l'utilisateur 4 a la permission "can\_post" sur l'objet GDC de numéro 3. Je pense que c'est le but qu'on devrait viser afin de "centraliser" la gestion des permissions. L'object\_type pourrait être le truc.\_meta.db\_table d'un objet.

#### #2 - 10 juillet 2014 16:37 - Frédéric Péters

Je porte cette discussion vers [#5108](#), pour laisser celui-ci à l'UI.

#### #3 - 11 juillet 2014 10:51 - Frédéric Péters

- Fichier 0001-ui-to-manage-access-rights-5124.patch ajouté

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

Patch qui ajoute les vues pour créer/lister/éditer/supprimer les ApiUser, et puis pour gdc une section sécurité permettant d'attacher ou retirer un apiuser.

#### #4 - 17 juillet 2014 15:30 - Frédéric Péters

- Fichier 0001-ui-to-manage-access-rights-5124.patch ajouté

J'ai rebasé pour avoir celui-ci avant le [#5122](#).

#### #5 - 17 juillet 2014 16:34 - Frédéric Péters

- Fichier 0001-ui-to-manage-access-rights-5124.patch ajouté

Thomas m'a fait remarquer qu'il restait une ligne dans gdc/urls.py qui ne servait à rien.

## #6 - 18 juillet 2014 11:06 - Frédéric Péters

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

Poussé avec une autre modification suggérée par Thomas, de limiter l'affichage du tableau des accès aux personnes autorisées.

```
commit 02ffe401ca426d20e5ea7376be75c057cc046920
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Thu Jul 10 13:10:06 2014 +0200
```

```
ui to manage access rights (#5124)
```

## #7 - 04 août 2018 12:32 - Benjamin Dauvergne

- Statut changé de *Résolu* (à déployer) à *Fermé*

### Fichiers

---

0001-ui-to-manage-access-rights-5124.patch	19,4 ko	11 juillet 2014	Frédéric Péters
0001-ui-to-manage-access-rights-5124.patch	25 ko	17 juillet 2014	Frédéric Péters
0001-ui-to-manage-access-rights-5124.patch	24,1 ko	17 juillet 2014	Frédéric Péters