Authentic 2 - Development #59664

Accorde ses propres permissions aux rôles : migré vers menu burger

14 décembre 2021 11:04 - Alexis Mathias

Statut:	Fermé	**Début:**	14 décembre 2021
Priorité:	Normal	**Echéance:**	
Assigné à:	Valentin Deniaud	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

En formation avec Stéphane à Roanne.

L'accès a "Accorde ses propres permissions aux rôles" a migré vers le menu burger et n'est pas accessible directement.
MAJ à venir apparemment (uniquement en recette)

Il est plus difficile à expliquer aux futurs utilisateurs.

Merci de voir si on peut revenir en arrière.

Demandes liées:

Lié à Authentic 2 - Development #57955: tableau affichant les membres "direct...	**Fermé**	**18 octobre 2021**

Révisions associées

Révision bd437dda - 04 janvier 2022 16:00 - Valentin Deniaud

manager: handle data-pk row attribute at view level (#59664)

Révision 84bc4a1a - 04 janvier 2022 16:03 - Valentin Deniaud

manager: add child roles in role members view (#59664)

Historique

#1 - 14 décembre 2021 11:05 - Frédéric Péters

> Merci de voir si on peut revenir en arrière.

Revenir en arrière non, aller en avant oui.

#2 - 14 décembre 2021 11:12 - Valentin Deniaud

- Lié à Development #57955: tableau affichant les membres "directs" d'un rôle ajouté

#3 - 14 décembre 2021 11:14 - Valentin Deniaud

Yep, développement arrivé hier en recette, il y a le temps d'en faire un truc qui convient à tout le monde avant l'arrivée en prod.

Plus que les discussions du ticket lié il faut lire/continuer celles de #56638.

#4 - 14 décembre 2021 11:15 - Frédéric Péters

On a désormais en effet un truc bizarre, pour ajouter un utiisateur dans le rôle, ça passe par la zone "Ajouter un utilisateur" qui est sous le tableau, alors que pour ajouter un rôle dans le rôle, ça passe par le menu burger.

Pour le premier, avec une sélection d'un utilisateur directement dans la page, pour l'autre avec un tableau des rôles où piocher ceux qui doivent être ajoutés.

Je serais pour avoir sous le tableau les deux, soit en les superposant,

```
Ajouter un utilisateur :  [ ............................. |v]  [Ajouter]
Ajouter un rôle :         [ ............................. |v]  [Ajouter]
```

Soit en ayant un champ de recherche combiné (ce qui aurait ma préférence),

```
Ajouter au rôle :  [ .(recherche usagers et rôles)......... |v]  [Ajouter]
```

#5 - 14 décembre 2021 11:55 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Soit en ayant un champ de recherche combiné (ce qui aurait ma préférence),
>
> [...]

C'est ce que j'avais évoqué dans #57955, ça a toujours ma préférence aussi :

> > je suppose qu'il faut penser ça dans une refonte de l'action "Ajouter un membre" qui dans un même élan devrait pouvoir traiter les rôles et les usagers.
>
> Option simple encore une fois, pas touche à cette action et à la place je déplace l'action pour ajouter un rôle comme membre dans le menu kebab, il me semble que c'est assez cohérent avec les discussions de #56638.

#6 - 14 décembre 2021 14:37 - Valentin Deniaud

Benjamin Dauvergne a écrit :

> Frédéric Péters a écrit :
>
> > Soit en ayant un champ de recherche combiné (ce qui aurait ma préférence),
>
> C'est ce que j'avais évoqué dans #57955, ça a toujours ma préférence aussi :

Et moi qui vais m'y coller j'ai regardé le code et ma préférence ça serait d'avoir deux champs distincts pour rôles et utilisateurs, mais bon puisqu'il y a consensus allons-y comme ça.

Techniquement ça serait OK de proposer un patch qui n'utilise pas django-select2 ? On perdrait deux choses, premièrement la mutualisation du code entre les champs, mais ce nouveau champ serait suffisamment différent pour que ce ne soit pas intéressant. Deuxièmement, le cache, là je n'arrive pas à savoir si ne pas en avoir poserait des vrais problèmes de perf ou pas du tout.

#7 - 14 décembre 2021 14:43 - Frédéric Péters

(perso totalement pour retirer django-select2, c'est une dépendance qui ne va pas poser problème pour bullseye mais que je voyais déjà amener des ennuis plus loin).

#8 - 15 décembre 2021 10:25 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

#9 - 15 décembre 2021 11:08 - Benjamin Dauvergne

Valentin Deniaud a écrit :

> Techniquement ça serait OK de proposer un patch qui n'utilise pas django-select2 ? On perdrait deux choses, premièrement la mutualisation du code entre les champs, mais ce nouveau champ serait suffisamment différent pour que ce ne soit pas intéressant.

Je n'ai rien contre, si ça peut donner une base pour migrer le reste tant mieux.

> Deuxièmement, le cache, là je n'arrive pas à savoir si ne pas en avoir poserait des vrais problèmes de perf ou pas du tout.

Je ne sais pas de quel cache tu parles; peux-tu expliciter ?

#10 - 15 décembre 2021 11:30 - Valentin Deniaud

Benjamin Dauvergne a écrit :

> Je ne sais pas de quel cache tu parles; peux-tu expliciter ?

https://django-select2.readthedocs.io/en/latest/django_select2.html#module-django_select2.cache, utilisé par le widget https://django-select2.readthedocs.io/en/latest/django_select2.html#django_select2.forms.HeavySelect2Mixin. Actuellement on a déjà bricolé ça pour

que ce ne soit plus l'objet queryset qui soit mis en cache mais juste le where de la requête. Mais tout bien réfléchi je ne sais même pas quel gain on a à faire ça.

#11 - 15 décembre 2021 15:40 - Benjamin Dauvergne

Valentin Deniaud a écrit :

> Benjamin Dauvergne a écrit :
>
> > Je ne sais pas de quel cache tu parles; peux-tu expliciter ?
>
> https://django-select2.readthedocs.io/en/latest/django_select2.html#module-django_select2.cache, utilisé par le widget https://django-select2.readthedocs.io/en/latest/django_select2.html#django_select2.forms.HeavySelect2Mixin. Actuellement on a déjà bricolé ça pour que ce ne soit plus l'objet queryset qui soit mis en cache mais juste le where de la requête. Mais tout bien réfléchi je ne sais même pas quel gain on a à faire ça.

J'ai l'impression que tu es embêté ici car on veut un sélecteur pouvant contenir deux modèles différents et que ça ne rentre pas dans le schéma de django-select2 : 1 field = 1 widget = 1 queryset = 1 modèle. Clairement ça ne sert à rien de reproduire quoi que ce soit de django-select2, il faut un champ spécifique qui ne peut pas être un ModelField (il faut passer l'information modèle + pk) et une vue AJAX spécifique pour renvoyer les réponses, il n'y a pas trop le choix, il n'y a aucun cache nécessaire. Pour moi la demande de virer django-select2 c'est forcément la demande d'avoir autant de vue AJAX que de champs select2 ou presque. Ce qui était sympa avec django-select2 c'est qu'une seule suffisait et ça marchait tout seul.

#12 - 21 décembre 2021 11:53 - Valentin Deniaud

- Fichier 0001-manager-handle-data-pk-row-attribute-at-view-level-5.patch ajouté

- Fichier 0002-manager-add-child-roles-in-role-members-view-59664.patch ajouté

- Tracker changé de Support à Development

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Voilà bon pour relecture mais surtout à tester en local.

J'ai gardé une petite dépendance select2, le widget, pour ne pas avoir à me farcir de html ni de js mais on pourra facilement prendre ça à notre charge dans un second temps.

En plus de l'ajout de rôles via le champ select2, il fallait aussi gérer la suppression via le tableau.

#13 - 04 janvier 2022 14:51 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

- 0001: nickel
- 0002: beau bordel, mais j'ai testé ça marche.

#14 - 04 janvier 2022 16:11 - Valentin Deniaud

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

```
commit 84bc4a1a56b47e0e4e5cf426c090f059bb41f27c
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date:   Tue Jan 4 16:00:24 2022 +0100

    manager: add child roles in role members view (#59664)

commit bd437ddafb8807d4b788b8f054d9491554a4450b
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date:   Mon Dec 20 16:40:44 2021 +0100

    manager: handle data-pk row attribute at view level (#59664)
```

#16 - 04 janvier 2022 16:57 - Pierre Cros

Chic mais il y a toujours l'entrée « Ajouter un rôle comme membre » dans le menu Burger, je pense qu'on pourrait la supprimer.

#17 - 04 janvier 2022 17:16 - Frédéric Péters

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

#18 - 04 janvier 2022 17:20 - Stéphane Laget

Cela change bcp de nos habitudes.
Peut-être que "Voir tous les membres" devrait être coché par défaut.

#19 - 04 janvier 2022 17:40 - Frédéric Péters

- Fichier Screenshot 2022-01-04 at 17-22-34 Authentic - Agent base de connaissance.png ajouté

En parallèle à #60271, pour répondre sur le lien dans le menu burger, j'ai cette proposition alternative, qui déplacerait le lien vers cette gestion "en masse" dans la page, sans être totalement convaincu. (je n'ai pas retrouvé dans les tickets les moments où cette possibilité de gestion en masse était notée comme utile).

#20 - 04 janvier 2022 17:50 - Pierre Cros

Pour moi il n'est même pas nécessaire d'ajouter le lien que propose Fred et qui rend le truc plus délicat à comprendre.

On tolère le fait que l'ajout se fasse nécessairement un par un pour les utilisateurs, on peut le tolérer pour les rôles aussi (l'ajout de plusieurs utilisateurs à un rôle est certainement un cas d'usage plus fréquent que l'ajout de plusieurs rôles à un rôle).

Si on veut prévoir quelque chose pour l'ajout de plusieurs rôles/utilisateurs, je suis d'avis qu'il faut traiter ça ensemble et dans un autre ticket.

#21 - 06 mars 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-manager-handle-data-pk-row-attribute-at-view-level-5.patch	7,27 ko	21 décembre 2021	Valentin Deniaud
0002-manager-add-child-roles-in-role-members-view-59664.patch	29,7 ko	21 décembre 2021	Valentin Deniaud
Screenshot 2022-01-04 at 17-22-34 Authentic - Agent base de connaissance.png	35,6 ko	04 janvier 2022	Frédéric Péters