

Combo - Development #56188

permettre d'ouvrir l'édition d'une page ou d'une hiérarchie de pages à un rôle

16 août 2021 16:26 - Frédéric Péters

| | | | |
|--|-----------------|----------------------|--------------|
| Statut: | Fermé | Début: | 16 août 2021 |
| Priorité: | Normal | Echéance: | |
| Assigné à: | Frédéric Péters | % réalisé: | 0% |
| Catégorie: | | Temps estimé: | 0:00 heure |
| Version cible: | | Planning: | Non |
| Patch proposed: | Oui | | |
| Description | | | |
| Aujourd'hui l'édition des pages est accessible uniquement aux admins, il s'agirait de permettre une ouverture de l'édition de certaines pages à un rôle; par exemple sur un portail agents on pourrait avoir une section "base de connaissances" où tout le monde pourrait ajouter/éditer les pages. | | | |

Révisions associées

Révision af5caa19 - 10 septembre 2021 10:11 - Frédéric Péters

general: attach edit_role and subpages_edit_role attributes to Page (#56188)

Révision 69425c62 - 10 septembre 2021 11:19 - Frédéric Péters

general: handle /manage/ access to users with page edit roles (#56188)

Historique

#1 - 27 août 2021 14:25 - Frédéric Péters

- Fichier 0001-general-attach-edit_role-and-subpages_edit_role-attr.patch ajouté
- Fichier 0002-general-handle-manage-access-to-users-with-page-edit.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

0001 qui fait l'ajout des attributs (edit_role et subpages_edit_role) et la page pour les positionner. 0002 qui fait le taf d'exploitation.

0002 est un peu chargé par l'ajout d'un ManagedPageMixin de contrôle d'accès qui a derrière demandé à modifier des vues qui étaient sous forme de fonctions en classes, ex:

```
-def page_search_cell_add_engine(request, page_pk, cell_reference, engine_slug):  
-     cell = get_object_or_404(SearchCell, pk=cell_reference.split('-')[1], page=page_pk)  
+class PageSearchCellAddEngine(ManagedPageMixin, View):
```

ça amène un gros diff mais la réalité est juste un changement d'indentation.

edit_role permet de donner à un rôle l'autorisation d'éditer la page en question.

subpages_edit_role permet de donner à un rôle l'autorisation d'éditer les pages filles, ainsi qu'en ajouter, c'est le cas "base de connaissance" où on veut permettre l'ajout de pages dans cette "rubrique" uniquement. Quand on est un agent avec ce type d'accès, la fenêtre d'ajout d'une page contient en plus un champ "page parente".

Il y a toute une série de fonctions qui ne sont pas ouvertes par cet accès restreint, comme la suppression de page ou la modification de l'ordre de celles-ci; c'est parfois juste une question d'appréciation, on pourrait par exemple débattre de la possibilité de supprimer, dans mon idée ("base de connaissances"), l'accès qu'on ouvre large aux agents pour contribuer de nouvelles pages, je trouve qu'on n'a pas envie de l'accompagner de la possibilité de supprimer les pages.

Côté code au final le seul truc majeur qui change, c'est Page.get_as_reordered_flat_hierarchy qui est refait de manière bien différente.

#2 - 27 août 2021 16:00 - Benjamin Dauvergne

Ça m'a l'air tout ok vu de loin.

Frédéric Péters a écrit :

Côté code au final le seul truc majeur qui change, c'est Page.get_as_reordered_flat_hierarchy qui est refait de manière bien différente.

Je ne sais pas si c'est voulu mais le fait de rebaser page.level au minimum dans un cas comme :

```
A1
  B1
    C1
      D1
  B2
    C2
```

où l'utilisateur a un droit d'édition sur les sous-pages de C1 et B2 ça va donner :

```
C1
  D1
B2
  C2
```

je pense qu'il faudrait rebaser le niveau de chaque branche indépendante à 0, un peu comme ça :

```
visible = set()
for page_path in page_hierarchy:
    page = page_path[-1][-1]
    if len(path_path) < 2:
        page.level = 0
    else:
        parent = page_path[-2][-1]
        if parent in visible:
            page.level = parent.level + 1
        else:
            page.level = 0
    visible.add(page)
```

#3 - 30 août 2021 15:44 - Frédéric Péters

- Fichier 0001-general-handle-manage-access-to-users-with-page-edit.patch ajouté

Voilà avec l'adaptation plus fine de la hiérarchie.

#4 - 07 septembre 2021 17:30 - Paul Marillonnet

- Une petite remarque sur la précédence entre page_edit_role et subpages_edit_role, à en lire le bout de code :

```
pages_hierarchy = [
    x
    for x in pages_hierarchy
    if x[-1][-1].edit_role_id in group_ids
    or any(y[-1].subpages_edit_role_id in group_ids for y in x[:-1])
]
```

on comprend que subpages_edit_role s'applique à toutes les pages descendantes dans la hiérarchie, même si un autre subpages_edit_role à été défini pour une page intermédiaire dans la hiérarchie.

À l'inverse, on pourrait s'attendre à ce que edit_role donne automatiquement le droit d'éditer des sous-pages mais il semblerait que non. Est-ce que c'est grave docteur ?

Est-ce qu'on ne va pas avoir le cas d'un utilisateur avec le rôle A permettant d'éditer une page, qui souhaite laisser aux utilisateurs ayant le rôle B le droit d'éditer les sous-pages de ladite page, et ceci sans pour autant posséder le rôle B mais en souhaitant quand même garder le droit d'édition des sous-pages ?

Formulé autrement, est-ce la précédence de subpages_edit_role d'une page aux subpages_edit_role(s) ses descendants ne doit pas aussi aussi s'appliquer pour page_edit_role de la page par rapport aux subpages_edit_role de ses descendants ?

- Rien à voir, mais je me pose la question des perms de

```
group_ids = [x.id for x in user.groups.all()]
if Page.objects.filter(
    Q(edit_role_id__in=group_ids) | Q(subpages_edit_role_id__in=group_ids)
).exists():
    return True
```

dans le décorateur sans trop savoir si je suis parano ou non. J'ai peur que ça tape un peu au vu de l'usage qui en est fait.

Est-ce qu'il n'y aurait pas simplement moyen de tenir à jour l'ensemble des rôles autorisant l'édition d'une quelconque page ou de ses sous-pages, et dans ce décorateur simplement tester l'intersection de cet ensemble avec celui des rôles de l'utilisateur ?

Bref juste des questions pour se mettre d'accord, ensuite je testerai en local.

#5 - 07 septembre 2021 17:37 - Frédéric Péters

1/ edit_role c'est pour une page la page en question. subpage_edit_role c'est pour les pages "enfants" de la page en question. Rien de plus rien de moins.

2/ on parle d'un appel par requête HTTP, vers le backoffice, ce n'est pas là qu'il y aura à travailler les perms en premier.

#6 - 07 septembre 2021 17:43 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Pour moi c'est ok.

#7 - 07 septembre 2021 17:45 - Paul Marillonnet

Frédéric Péters a écrit :

1/ edit_role c'est pour une page la page en question. subpage_edit_role c'est pour les pages "enfants" de la page en question. Rien de plus rien de moins.

Ok c'est le "rien de plus" pour lequel j'avais un doute, mais si c'est voulu ainsi alors très bien.

2/ on parle d'un appel par requête HTTP, vers le backoffice, ce n'est pas là qu'il y aura à travailler les perms en premier.

Ok.

#8 - 10 septembre 2021 12:21 - Frédéric Péters

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

```
commit 69425c62003a8d4e2bf4810ef03836bcf424279a
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Mon Aug 9 15:55:37 2021 +0200
```

```
general: handle /manage/ access to users with page edit roles (#56188)
```

```
commit af5caal99a977a3f7f1a0973613a273436bcb77c
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Mon Aug 9 15:39:30 2021 +0200
```

```
general: attach edit_role and subpages_edit_role attributes to Page (#56188)
```

#9 - 17 septembre 2021 16:17 - Frédéric Péters

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

Fichiers

| Fichier | Taille | Date | Auteur |
|---|---------|--------------|-----------------|
| 0001-general-attach-edit_role-and-subpages_edit_role-attr.patch | 7,14 ko | 27 août 2021 | Frédéric Péters |
| 0002-general-handle-manage-access-to-users-with-page-edit.patch | 50,4 ko | 27 août 2021 | Frédéric Péters |
| 0001-general-handle-manage-access-to-users-with-page-edit.patch | 51,6 ko | 30 août 2021 | Frédéric Péters |