

Authentic 2 - Development #50054

journal, filtrer sur un type d'événement

11 janvier 2021 21:46 - Frédéric Péters

Statut:	Fermé	Début:	11 janvier 2021
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Pour déboguer l'accès à un compte j'ai voulu filtrer le tableau des événements pour juste voir les "changement de mot de passe par un administrateur" et je n'ai pas pu. (j'ai simplement essayé de taper "changement de mot de passe par un administrateur" dans la zone de recherche).			
Je ne suis pas sûr que juste filtrer sur le type précis soit suffisant, plus tard j'ai voulu filtrer sur "demande de réinitialisation du mot de passe" et ça me serait bien allé que ce filtrage attrape aussi les "demande de réinitialisation du mot de passe par un administrateur".			

Révisions associées

Révision 91a190cd - 01 avril 2021 17:23 - Valentin Deniaud

manager: easier journal filtering by event types (#50054)

Historique

#1 - 23 février 2021 10:39 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

#2 - 23 février 2021 11:57 - Valentin Deniaud

- Fichier 0002-manager-filter-journal-entries-by-event-type-50054.patch ajouté
- Fichier 0001-journal-remove-typo-hiding-search-by-event-type-doc-.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Bon j'ai codé mon affaire (0002) puis en allant écrire les tests je vois que c'était déjà possible mais pas indiqué à cause d'une typo (0001).

Je pense que mon 0002 vaut le coup quand même parce qu'actuellement on ne peut rechercher qu'en connaissant les slugs, ce qui est OK pour nous mais moins pour les agents, mais je laisse un relecteur se prononcer avant d'écrire les tests.

#3 - 24 février 2021 23:24 - Benjamin Dauvergne

Je veux bien un screenshot pour voir ce que ça donne, mais déjà il y a un bout que je ne comprends pas :

```
self.fields['event_types'].choices = sorted(
    self.fields['event_types'].choices, key=lambda x: unicodedata.normalize('NFKD', x[1])
)
```

C'est quoi le souci avec le choice de base ?

#4 - 25 février 2021 08:59 - Valentin Deniaud

- Fichier out.ogv ajouté

Benjamin Dauvergne a écrit :

C'est quoi le souci avec le choice de base ?

Il est trié par slug, or ici on affiche le nom traduit, et donc on a l'impression d'un tri aléatoire sans ça (et il faut normaliser pour les événements genre « échec d'authentification »).

#5 - 26 février 2021 15:52 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Benjamin Dauvergne a écrit :

C'est quoi le souci avec le choice de base ?

Il est trié par slug, or ici on affiche le nom traduit, et donc on a l'impression d'un tri aléatoire sans ça (et il faut normaliser pour les évènements genre « échec d'authentification »).

C'est grandement un hack, ça ne march pas sur les majuscules ni sur la lettre suivante (ici ed classé avant éc):

```
>>> sorted(['éc', 'b', 'd', 'É', 'b', 'd', 'ed', 'b', 'd', 'E'], key=lambda s: unicodedata.normalize('NFKD', s))
['E', 'É', 'b', 'b', 'b', 'd', 'd', 'd', 'ed', 'éc']
```

Il faut appliquer les règles de collation pour la locale française pour faire bien les choses, c'est naturellement fait par postgresql quand on trie en base et que la bonne locale est posée, coté python ça peut-être via locale.strxfrm mais ça nécessite de bouger la locale globale via locale.setlocale(LC_ALL, 'fr_FR'). En moins global tu peux utiliser python3-icu qui implémente une API de locale non globale.

Mais pour pas trop te faire chier tu peux aussi recopier `wcs.qommon.misc.simplify()` qui te donnera certainement à peu de chose près le même comportement :

```
def simplify(s, space='-'):
    if s is None:
        return ''
    if not isinstance(s, six.text_type):
        if get_publisher() and get_publisher().site_charset:
            s = force_text('%s' % s, get_publisher().site_charset, errors='ignore')
        else:
            s = force_text('%s' % s, 'iso-8859-1', errors='ignore')
    s = force_text(unicodedata.normalize('NFKD', s).encode('ascii', 'ignore'))
    s = re.sub(r'^\w\s\'' % s, '', s).strip().lower()
    s = re.sub(r'\s\'' % s, space, s)
    return s
```

--

Concernant la vidéo je ne suis pas fan de la popup gratuite, en plus on n'a pas de retour visible sur l'évènement actuellement filtré. Ne pourrait-t-on utiliser une simple liste à choix ?

#6 - 01 mars 2021 14:59 - Valentin Deniaud

Benjamin Dauvergne a écrit :

Mais pour pas trop te faire chier tu peux aussi recopier `wcs.qommon.misc.simplify()` qui te donnera certainement à peu de chose près le même comportement :

Ouep OK je vais faire ça.

Concernant la vidéo je ne suis pas fan de la popup gratuite, en plus on n'a pas de retour visible sur l'évènement actuellement filtré. Ne pourrait-t-on utiliser une simple liste à choix ?

Fred dit dans la description que filtrer sur un seul type n'est pas suffisant. J'ajoute un retour visuel ?

#7 - 08 mars 2021 16:34 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Benjamin Dauvergne a écrit :

Mais pour pas trop te faire chier tu peux aussi recopier `wcs.qommon.misc.simplify()` qui te donnera certainement à peu de chose près le même comportement :

Ouep OK je vais faire ça.

Concernant la vidéo je ne suis pas fan de la popup gratuite, en plus on n'a pas de retour visible sur l'évènement actuellement filtré. Ne pourrait-t-on utiliser une simple liste à choix ?

Fred dit dans la description que filtrer sur un seul type n'est pas suffisant. J'ajoute un retour visuel ?

J'ai deux idées :

- une liste à choix multiple, mais je suis pas fan,
- utiliser des wildcard sur les noms des types d'évènements pour générer des méta types, ex. :
 - "manager.*" pour tout ce qui concerne le BO,
 - "*.password.*" pour tout ce qui concerne les mots de passe,
 - "*.user.*" pour tout ce qui concerne les utilisateurs.

La liste est là :

```
name = 'manager.role.administrator.role.addition'
name = 'manager.role.administrator.role.removal'
name = 'manager.role.administrator.user.addition'
name = 'manager.role.administrator.user.removal'
name = 'manager.role.creation'
name = 'manager.role.deletion'
name = 'manager.role.edit'
name = 'manager.role.inheritance.addition'
name = 'manager.role.inheritance.removal'
name = 'manager.role.membership.grant'
name = 'manager.role.membership.removal'
name = 'manager.user.activation'
name = 'manager.user.creation'
name = 'manager.user.deactivation'
name = 'manager.user.deletion'
name = 'manager.user.email.change.request'
name = 'manager.user.password.change'
name = 'manager.user.password.change.force'
name = 'manager.user.password.change.unforce'
name = 'manager.user.password.reset.request'
name = 'manager.user.profile.edit'
name = 'manager.user.sso.authorization.deletion'
name = 'user.deletion'
name = 'user.login'
name = 'user.login.failure'
name = 'user.logout'
name = 'user.password.change'
name = 'user.password.reset'
name = 'user.password.reset.failure'
name = 'user.password.reset.request'
name = 'user.profile.edit'
name = 'user.registration'
name = 'user.registration.request'
name = 'user.service.sso'
name = 'user.service.sso.authorization'
name = 'user.service.sso.unauthorization'
```

Dans un premier temps on peut juste couvrir le cas d'usage listé par Fred (on peut même cacher les types précis dans ce cas, ne garder que le méta-type qui groupe les deux) avec un wildcard "*.password.reset.*" et ajouter un booléen aux types d'évènements pour indiquer qu'ils ne doivent pas être visibles directement dans la liste. Avec une menu à deux niveaux ce serait joli :

Utilisateurs :

- * Connexion (user.login*)
- * Ré-init mot de passe (*.password.reset.*)

Rôles :

...

#8 - 10 mars 2021 16:36 - Valentin Deniaud

J'aime bien, sans faire de magie il y a moyen de s'en sortir de manière très simple avec genre dans le formulaire

```
+EVENT_TYPE_CHOICES = (
+    ('user.login', _('Login')),
+    ('password-reset', _('Password reset')),
+)
+
+ event_types = forms.ChoiceField(required=False, choices=EVENT_TYPE_CHOICES)
+
```

```
+ def clean_event_types(self):
+     return models.EventType.objects.filter(name__contains=self.cleaned_data['event_types'])
```

Je pars là dessus ? (par contre je visualise pas le menu à deux niveaux, si tu veux développer)

#9 - 23 mars 2021 14:23 - Benjamin Dauvergne

Ok. Groupes nommés : <https://docs.djangoproject.com/fr/3.1/ref/models/fields/#field-choices-named-groups> .

#10 - 24 mars 2021 12:10 - Valentin Deniaud

- Fichier 0001-manager-easier-journal-filtering-by-event-types-5005.patch ajouté

#11 - 24 mars 2021 12:33 - Benjamin Dauvergne

Je compte 6 types d'évènements qui ne sont couvert par rien de mieux que 'Tous':

```
user.service.sso
user.service.sso.authorization
user.service.sso.unauthorization
manager.user.creation
manager.user.email.change.request
manager.user.sso.authorization.deletion
```

il faut peut-être leur trouver une place dans ton classement (si c'est chiant de trouver une clé unique je me dit que tu pourrais avoir un truc de ce genre :

```
...
    ('user.email,profile.edit', _('Profile change')),
...
    patterns = choice.split(',')
    qs_filter = reduce(Q.__or__, (Q(name__contains=pattern) for pattern in patterns))
    return EventType.objects.filter(qs_filter)
```

) Je verrais plutôt des groupes fonctionnels encore plus regroupés et ciblés :

- Utilisateurs :
 - Connexion & SSO (login, creation, registration, sso, password change, password reset, front et back)
 - Mot de passe (password change, reset, front et back)
 - Modifications du profil (password change, email change, profile edit, en back ou front)
- Backoffice
 - Tout
 - Gestion des rôles (je ne suis pas sûr que ce soit utile d'avoir plus fin, on peut toujours en ajouter plus tard)
 - Gestion des utilisateurs

À voir ce que les CPTs qui seraient amené à regarder dans le journal préfèrent aussi (si t'en as un ou deux autour de toi).

#12 - 31 mars 2021 11:15 - Valentin Deniaud

Benjamin Dauvergne a écrit :

```
il faut peut-être leur trouver une place dans ton classement (si c'est chiant de trouver une clé unique je me dit que tu pourrais avoir un truc de ce genre :
[...])
```

Oui OK pour intégrer ce bout de code.

À voir ce que les CPTs qui seraient amené à regarder dans le journal préfèrent aussi

Je crois qu'ils s'en fichent :/

Je verrais plutôt des groupes fonctionnels encore plus regroupés et ciblés :

OK mais autant que tu écrives ici le EVENT_TYPE_CHOICES de tes rêves, plutôt que j'essaye de deviner dans quel groupe tu imagines chaque type évènement ?

#13 - 31 mars 2021 12:12 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Je verrais plutôt des groupes fonctionnels encore plus regroupés et ciblés :

OK mais autant que tu écrives ici le EVENT_TYPE_CHOICES de tes rêves, plutôt que j'essaye de deviner dans quel groupe tu imagines chaque type événement ?

Modulo les autorisations de SSO que je vais rajouter et les motifs que je vais préciser celui que j'ai donné couvre tout il me semble

```
* Tout
* Utilisateurs :
** Connexion & SSO (*.login.*, *.user.creation, *.user.registration*, *sso*, *.password.*)
** Mot de passe (*.password.*)
** Modifications du profil (*.password.*, manager.user.*activation, manager.user.(password,profile,email).*, ^
user.deletion, ^user.profile.*)
* Backoffice :
** Tout (manager.*)
** Gestion des rôles (manager.user.*)
** Gestion des utilisateurs (manager.role.*)
```

#14 - 31 mars 2021 12:13 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#15 - 01 avril 2021 17:23 - Valentin Deniaud

- Fichier 0001-manager-easier-journal-filtering-by-event-types-5005.patch ajouté

- Statut changé de En cours à Solution proposée

#16 - 01 avril 2021 17:38 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

#17 - 01 avril 2021 17:49 - Valentin Deniaud

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

```
commit 91a190cda703e463ac4fb3ee59f153e17bfe4eb2
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Wed Mar 24 12:04:36 2021 +0100
```

```
manager: easier journal filtering by event types (#50054)
```

#18 - 02 avril 2021 18:16 - Frédéric Péters

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

Fichiers

0002-manager-filter-journal-entries-by-event-type-50054.patch	5,78 ko	23 février 2021	Valentin Deniaud
0001-journal-remove-typo-hiding-search-by-event-type-doc-.patch	1,06 ko	23 février 2021	Valentin Deniaud
out.ogv	4,22 Mo	25 février 2021	Valentin Deniaud
0001-manager-easier-journal-filtering-by-event-types-5005.patch	4,57 ko	24 mars 2021	Valentin Deniaud
0001-manager-easier-journal-filtering-by-event-types-5005.patch	4,85 ko	01 avril 2021	Valentin Deniaud