

Authentic 2 - Development #49957

custom_user : dans `free_text_search` spécialiser en fonction du motif et utiliser sinon `find_duplicates(fullname=)` si on ne trouve pas mieux

07 janvier 2021 16:44 - Benjamin Dauvergne

Statut:	Fermé	Début:	07 janvier 2021
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Il faut spécialiser la recherche "libre", si c'est un email on cherche via email, si c'est un numéro de téléphone on cherche par numéro de téléphone sinon par trigramme sur le nom. Les combinaisons ne servent à rien.			
Demandes liées:			
Lié à Authentic 2 - Development #49899: /api/users/: autre tri par défaut			Fermé 05 janvier 2021

Révisions associées

Révision 3cb60a41 - 19 janvier 2021 15:21 - Benjamin Dauvergne

custom_user: index User.username and User.email (#49957)

Révision c98b0f23 - 19 janvier 2021 15:21 - Benjamin Dauvergne

authentic2: add full text search to AttributeValue (#49957)

Révision f4908a01 - 19 janvier 2021 15:21 - Benjamin Dauvergne

tests: use pytest style (#49957)

Révision ab663853 - 19 janvier 2021 15:21 - Benjamin Dauvergne

custom_user: specialize free_text_search for common search terms (#49957)

Historique

#1 - 07 janvier 2021 16:44 - Benjamin Dauvergne

- Tracker changé de Support à Development

#3 - 07 janvier 2021 16:46 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#4 - 08 janvier 2021 09:47 - Mikaël Ates (de retour le 29 avril)

- Lié à Development #49899: /api/users/: autre tri par défaut ajouté

#5 - 08 janvier 2021 12:09 - Benjamin Dauvergne

Désormais le tri pré-défini ne pas forcément pris en compte lorsque "q" est utilisé (idem dans le manager pour la recherche libre) :

- si un uuid, un téléphone, un nom d'utilisateur ou un email est donné, on utilise le trie par défaut
- si on passe en recherche par trigramme via `find_duplicates` alors on trie par le pourcentage de correspondance et rien d'autre
- si aucun des ces recherches spécifiques ne retourne de résultat on retourne sur l'ancien code (on cherche partout dans tous les attributs) et on trie par défaut

#6 - 11 janvier 2021 10:07 - Benjamin Dauvergne

- Fichier 0001-custom_user-specialize-free_text_search-for-common-s.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

J'ai conservé l'ancienne implémentation en dernier recours, des endroits peuvent en dépendre (identifiant RSA ou que sais-je).

#7 - 11 janvier 2021 10:34 - Benjamin Dauvergne

- Fichier 0001-custom_user-specialize-free_text_search-for-common-s.patch ajouté

avec le module manquant.

#8 - 11 janvier 2021 11:38 - Benjamin Dauvergne

- Fichier 0002-custom_user-specialize-free_text_search-for-common-s.patch ajouté

- Fichier 0001-tests-use-pytest-style.patch ajouté

et plus de tests.

#9 - 11 janvier 2021 12:30 - Valentin Deniaud

Pour ce qui est de l'email et de la date de naissance pas de pb, par contre la suite pose question, quelqu'un qui habite rue Deshayes va sortir dans la recherche jusqu'à ce qu'un Mr Deshayes se crée un compte, c'est un comportement bizarre.

En fait pour email ou date on a une manière suffisamment fiable pour déterminer ce que l'agent recherche, en revanche on ne sait pas distinguer un nom d'une adresse ou autre attribut texte. Pour moi ce serait mieux que find_duplicates intervienne dans free_text_search_complex plutôt qu'à la place (avec éventuellement en bonus un tri qui fasse ressortir les gens retournés par find_duplicates avant ceux dont un attribut matche).

#10 - 11 janvier 2021 14:25 - Benjamin Dauvergne

Valentin Deniaud a écrit :

En fait pour email ou date on a une manière suffisamment fiable pour déterminer ce que l'agent recherche, en revanche on ne sait pas distinguer un nom d'une adresse ou autre attribut texte. Pour moi ce serait mieux que find_duplicates intervienne dans free_text_search_complex plutôt qu'à la place (avec éventuellement en bonus un tri qui fasse ressortir les gens retournés par find_duplicates avant ceux dont un attribut matche).

Je ne sais pas faire cela, on peut trier par distance et rechercher par trigramme ou chercher librement dans tous les attributs, on ne peut pas faire les deux. Je serai plutôt pour abandonner la recherche libre sur les attributs dans ce cas là, en vrai on a nulle part de recherche sur l'adresse. Les gens recherchent par nom , téléphone ou email.

#11 - 11 janvier 2021 14:50 - Valentin Deniaud

Benjamin Dauvergne a écrit :

Je serai plutôt pour abandonner la recherche libre sur les attributs dans ce cas là, en vrai on a nulle part de recherche sur l'adresse. Les gens recherchent par nom , téléphone ou email.

Si ça ne sert à rien c'est bien de l'enlever, mais tu écrivais

J'ai conservé l'ancienne implémentation en dernier recours, des endroits peuvent en dépendre (identifiant RSA ou que sais-je).

Au pire un setting qui désactive le recours à free_text_search_complex par défaut, pour pouvoir le remettre vite si quelqu'un crie, et l'enlever sereinement au cycle d'après sinon ?

#12 - 11 janvier 2021 14:59 - Benjamin Dauvergne

- Fichier 0002-custom_user-specialize-free_text_search-for-common-s.patch ajouté

- Fichier 0001-tests-use-pytest-style.patch ajouté

C'est pas top, mais j'ai ajouté les autres matchs via les attributs sur le coté avec dist=1.0 et tri par défaut par nom puis prénom.

PS: j'ai aussi viré la recherche par email, first_name et last_name dans legacy qui faisaient doublon.

#13 - 11 janvier 2021 17:18 - Valentin Deniaud

(c'est rouge)

#14 - 11 janvier 2021 17:49 - Benjamin Dauvergne

- Fichier 0002-custom_user-specialize-free_text_search-for-common-s.patch ajouté

- Fichier 0001-tests-use-pytest-style.patch ajouté

Si l'un des queryset est "distinct()" l'autre doit l'être aussi.

#15 - 12 janvier 2021 10:17 - Valentin Deniaud

Malheureusement l'union qs1 | qs2 n'utilise plus l'index trigram, donc cette approche n'est pas la bonne...

(j'ai vu ça en utilisant la méthode explain(), puis pour confirmer j'ai utilisé la fixture large_userbase et rajouté 100 itérations dans test_fts_legacy_and_trigram, résultat j'ai un run à 3s avec juste qs1, un autre à 70s avec l'union)

#16 - 16 janvier 2021 19:18 - Benjamin Dauvergne

- Fichier 0004-custom_user-specialize-free_text_search-for-common-s.patch ajouté
- Fichier 0003-tests-use-pytest-style-49957.patch ajouté
- Fichier 0001-custom_user-index-User.username-and-User.email-49957.patch ajouté
- Fichier 0002-authentic2-add-full-text-search-to-AttributeValue-49.patch ajouté

La source des lenteurs venaient de ce que les recherches autres que par trigramme n'utilisaient pas d'index :

- j'ai ajouté un index UPPER sur username et email, pour pouvoir faire des recherches exactes insensibles à la casse
- j'ai ajouté un index FTS sur AttributeValue.content

Postgresql a du mal si le where est trop complexe : il devrait faire 4 index scan sur les 4 indexes, trigramme, username, email et fts mais il n'arrive pas à trouver ce plan tout seul alors je fais la jointure à la main, sachant qu'à la fin c'est toujours le match par trigramme qui compte pour le tri.

#17 - 19 janvier 2021 14:50 - Valentin Deniaud

- Statut changé de Solution proposée à Solution validée

Très bien, juste peut-être une méthode update_search_vectors() superflue puisque appelée nul part.

#18 - 19 janvier 2021 15:21 - Benjamin Dauvergne

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

```
commit ab6638531561df066271eb3200bdda6e7aa1468a
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Fri Jan 8 12:00:46 2021 +0100
```

```
custom_user: specialize free_text_search for common search terms (#49957)
```

```
commit f4908a01f4fc03c70c34c710b1585277aee454f0
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Mon Jan 11 10:59:38 2021 +0100
```

```
tests: use pytest style (#49957)
```

```
commit c98b0f2347e1087731ed4903bd845a1bef0005af
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Sat Jan 16 15:33:45 2021 +0100
```

```
authentic2: add full text search to AttributeValue (#49957)
```

```
commit 3cb60a412fc8a258470872c011237200446f4963
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Sat Jan 16 14:18:12 2021 +0100
```

```
custom_user: index User.username and User.email (#49957)
```

#19 - 22 janvier 2021 13:16 - Frédéric Péters

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

Fichiers

0001-custom_user-specialize-free_text_search-for-common-s.patch	4,12 ko	11 janvier 2021	Benjamin Dauvergne
0001-custom_user-specialize-free_text_search-for-common-s.patch	5,58 ko	11 janvier 2021	Benjamin Dauvergne
0002-custom_user-specialize-free_text_search-for-common-s.patch	7,87 ko	11 janvier 2021	Benjamin Dauvergne
0001-tests-use-pytest-style.patch	4,32 ko	11 janvier 2021	Benjamin Dauvergne
0002-custom_user-specialize-free_text_search-for-common-s.patch	9,54 ko	11 janvier 2021	Benjamin Dauvergne
0001-tests-use-pytest-style.patch	4,32 ko	11 janvier 2021	Benjamin Dauvergne
0002-custom_user-specialize-free_text_search-for-common-s.patch	9,6 ko	11 janvier 2021	Benjamin Dauvergne

0001-tests-use-pytest-style.patch	4,32 ko	11 janvier 2021	Benjamin Dauvergne
0004-custom_user-specialize-free_text_search-for-common-s.patch	11,2 ko	16 janvier 2021	Benjamin Dauvergne
0003-tests-use-pytest-style-49957.patch	4,33 ko	16 janvier 2021	Benjamin Dauvergne
0001-custom_user-index-User.username-and-User.email-49957.patch	1,94 ko	16 janvier 2021	Benjamin Dauvergne
0002-authentic2-add-full-text-search-to-AttributeValue-49.patch	5,61 ko	16 janvier 2021	Benjamin Dauvergne