

Authentic 2 - Development #22250

api OU : calcul automatique du slug quand il est absent

04 mars 2018 12:26 - Frédéric Péters

Statut:	Fermé	Début:	04 mars 2018
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Aujourd'hui il est demandé à l'appelant de l'API de spécifier un slug, ce n'est pas le cas via l'UI, l'API pourrait pareillement le déterminer toute seule.			

Révisions associées

Révision f41c11d2 - 17 septembre 2020 11:46 - Paul Marillonnet

api_views: provide a default slug for ous (#22250)

Historique

#1 - 04 avril 2018 10:38 - Paul Marillonnet

Même question que pour [#22251](#) : sur quoi se base-t-on pour déduire le slug ? Le nom de l'OU ?

#2 - 04 avril 2018 12:24 - Benjamin Dauvergne

Oui, slugify(ou.name).

#3 - 14 septembre 2020 17:28 - Paul Marillonnet

- Fichier 0001-api_views-provide-a-default-slug-for-ous-22250.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Comme ça, au plus simple possible, je pense.

#4 - 14 septembre 2020 17:31 - Paul Marillonnet

- Fichier 0001-api_views-provide-a-default-slug-for-ous-22250.patch ajouté

(Un assert en plus, sait-on jamais.)

#5 - 14 septembre 2020 17:52 - Paul Marillonnet

(Et pour info, c'est documenté ici <https://www.django-rest-framework.org/api-guide/fields/#default>.)

#6 - 15 septembre 2020 09:26 - Frédéric Péters

Mais s'il y a deux appels, le second plante parce que le slug est le même ?

#7 - 15 septembre 2020 09:54 - Benjamin Dauvergne

Du même avis que Fred, mais que le slug soit fourni ou calculer si les paramètre sont identiques, il faut juste renvoyer l'instance existante, je dirai qu'il faut faire un get_or_create() avec le slug seul, mettre à jour le nom si nécessaire et renvoyer l'instance trouvée. Ça ne peut pas se faire au niveau du serializer mais il faudrait plutôt surcharger ModelSerializer.create().

#8 - 15 septembre 2020 10:00 - Paul Marillonnet

Frédéric Péters a écrit :

Mais s'il y a deux appels, le second plante parce que le slug est le même ?

S'il y a deux appels avec le même nom, on aura déjà un souci d'unicité, indépendamment du slug :

```
E      Bad response: 400 Bad Request (not 200 OK or 3xx redirect for http://testserver/api/ous/)
E      b'{"result":0,"errors":{"__all__":["The fields name must make a unique set.,"The fields slug must
make a unique set."]}}'
```

#9 - 15 septembre 2020 10:03 - Paul Marillonnet

- Statut changé de Solution proposée à Information nécessaire

Benjamin Dauvergne a écrit :

Du même avis que Fred, mais que le slug soit fourni ou calculer si les paramètres sont identiques, il faut juste renvoyer l'instance existante, je dirai qu'il faut faire un `get_or_create()` avec le slug seul, mettre à jour le nom si nécessaire et renvoyer l'instance trouvée. Ça ne peut pas se faire au niveau du `serializer` mais il faudrait plutôt surcharger `ModelSerializer.create()`.

Je ne suis pas convaincu par le comportement `get_or_create` par défaut sur le POST. Pour moi si l'instance existe déjà il faut renvoyer un 40X, non ? Dans l'API de rôles, par exemple, on n'a ces comportements `"*_or_create"` qu'en précisant des paramètres de `querystring` spécifiques.

#10 - 15 septembre 2020 10:45 - Frédéric Péters

S'il y a deux appels avec le même nom, on aura déjà un souci d'unicité, indépendamment du slug :

Ok ajoute une modification du nom entre temps (ou utiliser deux noms différents qui donnent le même slug) et voilà tu as l'erreur dont je voulais parler. Ce que je veux dire c'est que via l'UI, où on n'entre pas de slug, on peut faire "créer A", "renommer A en B", "créer A", et le slug de ce second A est créé différent de celui de l'A initial. (a1 vs a)

#11 - 15 septembre 2020 10:49 - Benjamin Dauvergne

Paul Marillonnet a écrit :

Benjamin Dauvergne a écrit :

Du même avis que Fred, mais que le slug soit fourni ou calculer si les paramètres sont identiques, il faut juste renvoyer l'instance existante, je dirai qu'il faut faire un `get_or_create()` avec le slug seul, mettre à jour le nom si nécessaire et renvoyer l'instance trouvée. Ça ne peut pas se faire au niveau du `serializer` mais il faudrait plutôt surcharger `ModelSerializer.create()`.

Je ne suis pas convaincu par le comportement `get_or_create` par défaut sur le POST. Pour moi si l'instance existe déjà il faut renvoyer un 40X, non ?

Dans l'API de rôles, par exemple, on n'a ces comportements `"*_or_create"` qu'en précisant des paramètres de `querystring` spécifiques.

Il y a la théorie et la pratique :) En théorie si on était des puristes, on devrait détecter l'erreur et la gérer dans le workflow qui appelle ça, en pratique les workflows sont beaucoup plus simple si aucune erreur n'est possible dans le fonctionnement normal (i.e. si les appels sont idempotents, en cas d'erreur technique temporaire on a juste à répéter les appels et ça fonctionne). Le cas d'un nom très différent donnant un slug identique sont impossibles. Dans les cas où on doit vraiment vérifier si l'objet existe déjà (et j'ai l'impression que pour les OUs ça n'arrivera pas) il n'y a qu'à faire un `get` avant.

Maintenant si tu veux vraiment avoir les deux comportements sur un même endpoint il faut réutiliser le `GetOrCreateMixin` utilisé sur l'API des utilisateurs (mais ça complexifie, à la fois les appels et les implémentations).

#12 - 15 septembre 2020 10:50 - Benjamin Dauvergne

Frédéric Péters a écrit :

S'il y a deux appels avec le même nom, on aura déjà un souci d'unicité, indépendamment du slug :

Ok ajoute une modification du nom entre temps (ou utiliser deux noms différents qui donnent le même slug) et voilà tu as l'erreur dont je voulais parler. Ce que je veux dire c'est que via l'UI, où on n'entre pas de slug, on peut faire "créer A", "renommer A en B", "créer A", et le slug de ce second A est créé différent de celui de l'A initial. (a1 vs a)

Ce problème là plaide simplement pour exposer le champ slug (dans l'IHM), le slug est sensé être immuable mais pas de manière absolue, ce n'est pas un UUID, on doit pouvoir corriger un erreur temporaire de nommage à la main.

#13 - 15 septembre 2020 11:02 - Paul Marillonnet

Frédéric Péters a écrit :

Ok ajoute une modification du nom entre temps (ou utiliser deux noms différents qui donnent le même slug) et voilà tu as l'erreur dont je voulais parler. Ce que je veux dire c'est que via l'UI, où on n'entre pas de slug, on peut faire "créer A", "renommer A en B", "créer A", et le slug de ce second A est créé différent de celui de l'A initial. (a1 vs a)

Ok oui je n'avais pas pensé à ce comportement, qu'on pourrait reproduire dans l'API.

#14 - 15 septembre 2020 11:07 - Paul Marillonnet

- Statut changé de *Information nécessaire à En cours*

Benjamin Dauvergne a écrit :

Il y a la théorie et la pratique :) En théorie si on était des puristes, on devrait détecter l'erreur et la gérer dans le workflow qui appelle ça, en pratique les workflows sont beaucoup plus simple si aucune erreur n'est possible dans le fonctionnement normal (i.e. si les appels sont idempotents, en cas d'erreur technique temporaire on a juste à répéter les appels et ça fonctionne). Le cas d'un nom très différent donnant un slug identique sont impossibles. Dans les cas où on doit vraiment vérifier si l'objet existe déjà (et j'ai l'impression que pour les OUs ça n'arrivera pas) il n'y a qu'à faire un get avant.

Ok, j'entends l'argument maintenant. (À noter quand même qu'il n'y aura pas idempotence complète parce que les codes de retour HTTP, entre une création et une lecture, vont sans doute varier — i.e 201 vs 200).

Maintenant si tu veux vraiment avoir les deux comportements sur un même endpoint il faut réutiliser le GetOrCreateMixin utilisé sur l'API des utilisateurs (mais ça complexifie, à la fois les appels et les implémentations).

Je ne me rends pas compte de l'ajout de complexité (dans mes souvenirs le mixin dédié parvient bien à contenir la complexité de la chose). Je vais regarder ça.

#15 - 16 septembre 2020 08:53 - Paul Marillonnet

- Fichier *0001-api_views-provide-a-default-slug-for-ous-22250.patch* ajouté

- Statut changé de *En cours* à *Solution proposée*

Paul Marillonnet a écrit :

Je ne me rends pas compte de l'ajout de complexité (dans mes souvenirs le mixin dédié parvient bien à contenir la complexité de la chose). Je vais regarder ça.

Et bien oui, ou alors je loupe quelque chose ? Ça m'a tout l'air de fonctionner comme ça.

#16 - 17 septembre 2020 10:39 - Benjamin Dauvergne

Paul Marillonnet a écrit :

Je ne me rends pas compte de l'ajout de complexité (dans mes souvenirs le mixin dédié parvient bien à contenir la complexité de la chose). Je vais regarder ça.

Ça complexifie dans le sens où la personne qui fait l'appel doit penser qu'elle veut le comportement get-or-create et donc ajouter la clé ?get-or-create=slug dans la query-string et aussi qu'elle doit savoir que c'est sur le champ slug qu'il faut agir, ça fait beaucoup de détails techniques à transmettre via la documentation ou même simplement à apprendre aux CPFs.

#17 - 17 septembre 2020 10:41 - Benjamin Dauvergne

- Statut changé de *Solution proposée* à *Solution validée*

Benjamin Dauvergne a écrit :

Ça complexifie dans le sens où la personne qui fait l'appel doit penser qu'elle veut le comportement get-or-create et donc ajouter la clé ?get-or-create=slug dans la query-string et aussi qu'elle doit savoir que c'est sur le champ slug qu'il faut agir, ça fait beaucoup de détails techniques à transmettre via la documentation ou même simplement à apprendre aux CPFs.

Je ne suis pas fan, mais je valide sauf si d'autres trouvent aussi que c'est bien chiant pour rien de faire comme ça.

#19 - 17 septembre 2020 18:09 - Paul Marillonnet

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

commit f41c11d259371fb83d112a0fd09672238997ee3e
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date: Mon Sep 14 16:53:24 2020 +0200

```
api_views: provide a default slug for ous (#22250)
```

#20 - 18 septembre 2020 08:16 - Frédéric Péters

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

Fichiers

0001-api_views-provide-a-default-slug-for-ous-22250.patch	3,06 ko	14 septembre 2020	Paul Marillonnet
0001-api_views-provide-a-default-slug-for-ous-22250.patch	3,1 ko	14 septembre 2020	Paul Marillonnet
0001-api_views-provide-a-default-slug-for-ous-22250.patch	4,74 ko	16 septembre 2020	Paul Marillonnet