

Passerelle - Bug #5779

rendre les slugs uniques

21 octobre 2014 15:36 - Thomas Noël

Statut:	Rejeté	Début:	21 octobre 2014
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Non		
Description			
Actuellement, si on crée un connecter "gdc" appelé "truc", ça fait un slug "truc". Si on crée un second connecteur "truc", ça fait encore un slug "truc" et on arrive sur une erreur : MultipleObjectsReturned.			
Il faut rendre les slugs uniques par type de service.			
Demandes liées:			
Bloque Passerelle - Bug #5778: Permettre la modification des slugs		Fermé	21 octobre 2014
Bloqué par Passerelle - Bug #4696: Migration initiale depuis le script init.d		Rejeté	15 avril 2014

Historique

#1 - 21 octobre 2014 15:42 - Thomas Noël

- *Priorité changé de Normal à Haut*

Je monte la priorité parce que ça rend Passerelle inutilisable en cas de fausse manip, et que j'ai pas trop d'idée pour résoudre le soucis (donc oui c'est un peu un appel à l'aide)

#2 - 21 octobre 2014 16:04 - Benjamin Dauvergne

Voir l'option `unique_together`¹ dans Django.

¹<https://docs.djangoproject.com/en/dev/ref/models/options/#unique-together>

#3 - 21 octobre 2014 16:08 - Benjamin Dauvergne

Ok le type de service n'est pas un champ, c'est une table. Vous l'avez dans l'os, à moins de déplacer le champ slug du modèle parent au modèle enfant. Ça vous sert encore à quelque chose d'utiliser l'héritage ? J'ai l'impression qu'il n'y a rien de générique et dans ce cas vous pourriez migrer vers de l'héritage abstrait et rendre slug unique.

Comme on est pas dans un monde idéal je rendrais simplement le champ slug unique tout court, je ne vois pas bien la conséquence.

Autre possibilité c'est de tester ça dans le clean du formulaire d'édition, mais ce n'est pas 100% secure sans changer le niveau d'isolation de la base à `serialized`; dans votre cas ce ne serait pas gênant (dans une application où il y a beaucoup d'écriture ça fait des bugs étrange pour les utilisateurs voir calebasse).

#4 - 21 octobre 2014 16:55 - Thomas Noël

Bon, ça doit pouvoir se jouer avec `unique=True` dans le champ slug de `BaseResource`. Mais ça va impacter toutes les applications, il faut donc toutes les migrer (au sens "south").

Chemin d'upgrade à la main :

1. créer les migrations "initiales" pour les applis qui n'en ont pas
2. faire un `migrate --fake`
3. faire la mise à jour du slug avec `unique=True`
4. faire un `migrate`

Mais c'est un chemin impossible en mode "deb/rpm". Il faudrait plutôt avoir un script un peu sioux, à la sauce Python 1.7, c'est à dire :

- pour chaque application, lors du `init.d` :
 1. si la table n'existe pas, on `syncdb --migrate --fake` l'application
 2. si la table existe mais que y'a rien dans l'historique south, on `migrate-fake` le `0001_initial` puis on `migrate`.

#5 - 21 octobre 2014 16:59 - Frédéric Péters

Le bug pour les migrations initiales existe déjà, [#4696](#).

#6 - 11 mai 2015 11:08 - Frédéric Péters

- *Priorité changé de Haut à Normal*

#7 - 11 décembre 2017 22:47 - Frédéric Péters

- *Statut changé de Nouveau à Rejeté*

Doublon de [#13053](#) où on va vers un patch.