

Passerelle - Development #42008

opengis, récupération d'une feature sur base d'un identifiant

23 avril 2020 09:00 - Frédéric Péters

Statut:	Fermé	Début:	23 avril 2020
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
On a ?q= pour une recherche textuelle, parallèlement on pourrait avoir ?id= pour récupérer un geojson limité à la "feature" demandée.			
Ça demanderait dans la configuration de préciser la propriété correspond à l'identifiant mais plutôt que ça je me demande si on ne pourrait pas juste avoir un générique ?<propriété>=<value>, qui marcherait pour n'importe quelle propriété de FeatureCache::data.			

Révisions associées

Révision 45fe112a - 18 mai 2020 10:16 - Valentin Deniaud

utils: allow explicit flag on optional endpoint param (#42008)

Révision 5e0369d4 - 18 mai 2020 10:16 - Valentin Deniaud

opengis: filter custom query on any field (#42008)

Historique

#1 - 29 avril 2020 16:00 - Valentin Deniaud

Et je découvre qu'avec django-jsonfield, on ne peut pas faire de lookup sur une clé, ie `features.filter(data__property=value)` ne fonctionne pas. C'est bête, ce code est récent, rien n'empêchait de partir sur le `JSONField` dans `django.contrib.postgres` qui supporte très bien ce genre de chose.

Je tente d'ajouter une migration de l'un vers l'autre ? Il y a des précédents ?

#2 - 29 avril 2020 16:11 - Frédéric Péters

`FeatureCache.objects.all().filter(data__contains={'ville': 'Bron'})` fonctionne.

Mais oui on peut imaginer la migration, j'ai l'impression qu'en terme de stockage db ça ne change rien. (il y a eu des trucs de cet ordre dans `authentic`, je crois).

#3 - 29 avril 2020 16:37 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

Frédéric Péters a écrit :

```
FeatureCache.objects.all().filter(data__contains={'ville': 'Bron'}) fonctionne.
```

Effectivement merci, mais ça devient vite relou si on commence à imaginer `{'propriétés': {'ville': 'Bron'}}`, et en plus on pourrait s'éviter tout parsing du paramètre reçu. Donc je vais quand même tenter la migration.

#4 - 29 avril 2020 16:40 - Frédéric Péters

Tout à fait d'accord.

#5 - 29 avril 2020 18:38 - Valentin Deniaud

Bon j'ai vu le code dans a2, j'ai pris peur et j'ai essayé la solution simple (migration qui ajoute `new_data`, `RunPython new_data = data`, supprime `data` et renomme `new_data -> data`) et je me suis pris <https://github.com/adamchainz/django-jsonfield/issues/5> en pleine tête (on ne peut pas utiliser le field de postgres tant que le projet utilise `django-jsonfield`, ce jusqu'à la version 1.3 et on a comme contrainte `< 1.3`). À part à faire sauter la contrainte de version, je vais revenir au filtrage permis par `django-jsonfield`...

#6 - 29 avril 2020 22:00 - Frédéric Péters

tant que le projet utilise django-jsonfield

Mais on peut pas juste tout virer ça ? Il me semble qu'on ne fait vraiment rien de particulier, qu'il n'y aurait même pas besoin de migration (juste trafiquer une précédente pour faire croire qu'on a depuis toujours utilisé le jsonfield de django.contrib.postgres).

#7 - 05 mai 2020 11:20 - Valentin Deniaud

- Fichier 0001-utils-allow-explicit-flag-on-optional-endpoint-param.patch ajouté

- Fichier 0002-opengis-filter-custom-query-on-any-field-42008.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Et donc ce que ça donne avec [#42312](#).

#8 - 10 mai 2020 17:14 - Frédéric Péters

data__properties__code_insee=..., je préférerais zapper la partie data__properties, limiter la recherche au contenu de la clé properties, et dans la querystring se serait property__code_insee=... (oui, property, pas properties, parce qu'on filtre sur une seule). (j'hésite aussi à proposer property:code_insee plutôt que le double underscore).

```
if s.lower() == 'true':
    return True
if s.lower() == 'false':
    return False
```

Je ne ferais pas ça pour le moment, côté data.grandlyon il y a quantité de propriétés avec 'true' mis sous forme de chaîne (genre "acceshandi": "true").

Dans une évolution où on aurait besoin de filtrer sur base d'un autre type, on pourrait passer par un autre paramètre, genre bool_property__whatever=true.

#9 - 11 mai 2020 10:29 - Valentin Deniaud

- Statut changé de Solution proposée à En cours

Frédéric Péters a écrit :

data__properties__code_insee=..., je préférerais zapper la partie data__properties, limiter la recherche au contenu de la clé properties, et dans la querystring se serait property__code_insee=... (oui, property, pas properties, parce qu'on filtre sur une seule). (j'hésite aussi à proposer property:code_insee plutôt que le double underscore).

OK mais c'est dommage de perdre la généralité et le code trivial associé, la clé properties c'est un détail d'implément du serveur, le jour où on voudra supporter autre chose il faudra toucher le code et le complexifier.

Je ne ferais pas ça pour le moment, côté data.grandlyon il y a quantité de propriétés avec 'true' mis sous forme de chaîne (genre "acceshandi": "true").

Et ça continuait à marcher, les filtres avec les conversions viennent en plus et pas à la place, mais je retire c'est pas beau de toute façon.

#10 - 11 mai 2020 10:50 - Frédéric Péters

la clé properties c'est un détail d'implément du serveur

Ça fait partie de la définition du format geojson.

#11 - 11 mai 2020 11:50 - Valentin Deniaud

Frédéric Péters a écrit :

Ça fait partie de la définition du format geojson.

Dac je savais pas, ça me va mieux.

Valentin Deniaud a écrit :

Frédéric Péters a écrit :

Je ne ferais pas ça pour le moment, côté data.grandlyon il y a quantité de propriétés avec 'true' mis sous forme de chaîne (genre "accesshandl": "true").

Et ça continuait à marcher, les filtres avec les conversions viennent en plus et pas à la place, mais je retire c'est pas beau de toute façon.

Mais je garde bien la tentative de conversion des nombres ?

#12 - 11 mai 2020 11:54 - Frédéric Péters

Mais je garde bien la tentative de conversion des nombres ?

Je n'y avais même pas prêté attention; et je n'arrive même pas à voir dans

```
kwargs.update({k: self.cast_str(v) for k, v in kwargs.items() if self.cast_str(v) is not None})
```

que ça vient en plus et pas à la place; en soit je veux bien, y compris le true/false, et les nombres, mais avec un commentaire là-dessus pour noter l'intention, et un test ajouté pour le filtre sur une valeur numérique/booléenne.

#13 - 11 mai 2020 13:31 - Valentin Deniaud

- Fichier 0001-utils-allow-explicit-flag-on-optional-endpoint-param.patch ajouté
- Fichier 0002-opengis-filter-custom-query-on-any-field-42008.patch ajouté
- Statut changé de En cours à Solution proposée

Frédéric Péters a écrit :

que ça vient en plus et pas à la place

Hum c'est parce que ça ne marchait pas du tout, on a rien vu.

Correction et ajout du bout de test qui va bien.

#14 - 16 mai 2020 09:31 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

#15 - 18 mai 2020 10:16 - Valentin Deniaud

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

```
commit 2d59f4ef7f43767977fd7bf1efeb6613da770a
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Wed Apr 29 15:58:34 2020 +0200
```

```
opengis: filter custom query on any field (#42008)
```

```
commit b76f43ca8f41a4a8856f5fbd7f48801e457e68a4
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Tue May 5 11:12:14 2020 +0200
```

```
utils: allow explicit flag on optional endpoint param (#42008)
```

#16 - 19 mai 2020 11:17 - Frédéric Péters

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

Fichiers

0001-utils-allow-explicit-flag-on-optional-endpoint-param.patch	867 octets	05 mai 2020	Valentin Deniaud
0002-opengis-filter-custom-query-on-any-field-42008.patch	4,53 ko	05 mai 2020	Valentin Deniaud
0001-utils-allow-explicit-flag-on-optional-endpoint-param.patch	867 octets	11 mai 2020	Valentin Deniaud
0002-opengis-filter-custom-query-on-any-field-42008.patch	4,91 ko	11 mai 2020	Valentin Deniaud