

## Passerelle - Development #50212

### opendatasoft, ajouter un paramètre de filtres dans les requêtes

17 janvier 2021 11:05 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	17 janvier 2021
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Nicolas Roche	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
Pour avoir les établissements, on instancie un connecteur avec <a href="https://data.education.gouv.fr/">https://data.education.gouv.fr/</a> et on crée une requête pour préciser le jeu de données fr-en-annuaire-education.			
Je pense utile qu'on puisse également dans la requête définir d'éventuels filtres, pour par exemple créer une requête qui limiterait les résultats aux établissements d'une région, ce qui se passerait ainsi côté API :			
<a href="https://data.education.gouv.fr/api/records/1.0/search/?dataset=fr-en-annuaire-education&amp;refine.libelle_region=Nouvelle-Aquitaine">https://data.education.gouv.fr/api/records/1.0/search/?dataset=fr-en-annuaire-education&amp;refine.libelle_region=Nouvelle-Aquitaine</a>			

#### Révisions associées

##### Révision 9b88f614 - 27 août 2021 14:05 - Nicolas Roche

opendatasoft: correct tests (#50212)

##### Révision ecd314d7 - 27 août 2021 14:05 - Nicolas Roche

opendatasoft: use APIError into endpoint (#50212)

##### Révision cac06da8 - 27 août 2021 14:05 - Nicolas Roche

opendatasoft: add facet filters (#50212)

#### Historique

##### #1 - 18 janvier 2021 14:30 - Nicolas Roche

Il est écrit "d'éventuels filtres"

Vu <https://help.opendatasoft.com/apis/ods-search-v1/#dataset-search-api>, j'en déduis :

- choisir le type de filtrage
- avoir plusieurs filtres

par exemple, proposer un bouton "ajouter un filtre" qui ajoute des lignes telles que celle là ?

```
+-----+
|v| opérateur | facette: | libelle_region | value: | Nouvelle-Aquitaine |
+-----+
| refine      |
| exclude    |
+-----+
```

##### #2 - 06 avril 2021 09:27 - Frédéric Péters

par exemple, proposer un bouton "ajouter un filtre" qui ajoute des lignes telles que celle là ?

Je pensais plutôt rester similaire à ce qu'on a pour opengis, quelque chose d'assez brut, ajouter un filtre ça serait un paramètre "valeur du filtre" et dedans on écrirait "refine.libelle\_region=Nouvelle-Aquitaine" et voilà.

##### #3 - 09 avril 2021 12:10 - Nicolas Roche

- Fichier 0001-opendatasoft-add-facet-filters-50212.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

1. Je n'ai pas mis de validateur sur le nouveau champ car je n'ai pas trouvé quelle syntaxe pouvait faire planter `urlparse.parse_qs(filter_expression, strict_parsing=True)`.

2. En lui ajoutant un champs texte dédié, l'objet Query fonctionne bien : je passe son contenu en paramètre à `search` qui l'utilise lors du GET.

Mais si j'appelle directement `search`, alors je ne sais pas comment traiter ce "morceau d'url qui contient des '&'" reçu via un paramètre de l'URL.

un paramètre "valeur du filtre" et dedans on écrirait `"refine.libelle_region=Nouvelle-Aquitaine"` et voilà.

En fait ce n'est pas si simple parce qu'il s'agirait d'un paramètre pouvant contenir plusieurs filtres.

Voici un exemple d'expression qu'il me faut prévoir de passer à `search` comme paramètre de l'URL par l'utilisateur, en utilisant un pré-supposé paramètre `"filter_expression"` dédié dans le endpoint `search` :

```
filter_expression=refine.facetA=valueA&refine.facetB=valueB'
```

Pour contourner le problème des '&' (faute de mieux) ce patch utilise le kwarg pour passer les filtres.

#### #4 - 12 avril 2021 09:50 - Frédéric Péters

Je ne comprends pas vraiment ce que tu essaies d'exposer comme problème; oui un champ où on pourrait écrire

`filter_expression=refine.facetA=valueA&refine.facetB=valueB`, c'est le côté un peu galère de demander d'écrire ça qui t'ennuie ? (et je n'ai pas compris si quelque chose marchait ou pas, au final).

```
+from django.utils.six.moves.urllib import parse as urlparse
```

On se passe désormais de six mais ici surtout il est ajouté et pas utilisé du tout ?

#### #5 - 12 avril 2021 10:26 - Nicolas Roche

- Sujet changé de `opendatasoft`, ajouter un paramètre de filtres dans les requêtes à `opendatasoft`, ajoute un paramètre de filtres dans les requêtes

```
filter_expression=refine.facetA=valueA & refine.facetB=valueB
```

Le & fait que je n'arrive pas à passer ce paramètre au endpoint `search` via l'API HTTP (appelé directement via un appel WS, sans passer par l'objet Query).

Je n'ai pas trouvé de code ailleurs où l'on passerait un tel paramètre "composé".

Ici pour avoir un truc propre, je dirais qu'il faudrait ajouter un champ par filtre, mais je préfère poser la question parce-que là non-plus ce n'est pas fait comme ça ailleurs.

Dans l'immédiat, je propose de passer les paramètres décomposés dans le kwarg au endpoint `search`, parce que c'est ce qui me semble le plus simple :

```
filters = urlparse.parse_qs(self.filter_expression)
kwargs.update(filters)
```

(merci pour la remarque sur l'import que j'ai oublié de retirer)

#### #6 - 12 avril 2021 10:48 - Frédéric Péters

Ça ne me semble pas délirant de se dire qu'on décompose le bout de chaîne passé pour pouvoir l'exploiter.

#### #7 - 12 avril 2021 12:54 - Nicolas Roche

- Fichier `0001-opendatasoft-add-facet-filters-50212.patch` ajouté

Tant mieux si ça ne choque pas.

J'ai pas été bien clair (mais tu m'as compris) et je précise pour d'autres qui passeraient par là, que l'appel au endpoint `search` (avec des filtres) aurait cette tête :

```
$ curl 'https://passerelle.dev.publik.love/opendatasoft/test/search?dataset=world-heritage-unesco-list&refine.continent_fr=Europe+et+Amérique+du+nord&refine.country_fr=France&exclude.category=Cultural'
```

Et qu'il n'y pas moyen d'indiquer nativement dans l'IHM que l'on peut ajouter des paramètres `"[refine|exclude].{facette}={value}"` (je n'ai rien prévu dans ce patch pour l'indiquer).

## #8 - 12 avril 2021 12:58 - Frédéric Péters

que l'appel au endpoint search (avec des filtres) aurait cette tête :

J'allais passer sous silence cet effet de bord mais puisque tu en parles, je suis d'avis de poser les choses : il ne faut donc pas prendre ce raccourci, l'endpoint "q" devrait appeler une méthode qui n'est pas l'endpoint (et l'endpoint "search" appellerait le même).

## #9 - 12 avril 2021 15:46 - Nicolas Roche

- Statut changé de Solution proposée à Information nécessaire

- Assigné à mis à Nicolas Roche

l'endpoint "q" devrait appeler une méthode qui n'est pas l'endpoint

Ok, mais ça ne résoudra pas mon problème, j'en déduis donc que je me suis encore mal exprimé.

Lorsque j'appelle un endpoint, je n'arrive pas à passer en l'état un paramètre qui contient un '&' afin que ce paramètre soit ensuite utilisé pour passer plusieurs paramètres GET à la requête vers le serveur opendatasoft.

Pour l'objet Query, cela fonctionne parce que j'utilise le paramètre stocké en base dans le champ texte, mais si je voulais passer ce champs en paramètre au endpoint q, alors la ligne curl serait du même type que ci-dessus (là ce n'est même pas prévu).

J'ai l'impression qu'il n'y a pas de solution élégante pour gérer ce champs composé, et que le plus conventionnel serait de convenir d'un séparateur autre que '&' pour passer les filtres dans un unique paramètre.

## #10 - 12 avril 2021 16:40 - Frédéric Péters

- Sujet changé de opendatasoft, ajoute un paramètre de filtres dans les requêtes à opendatasoft, ajouter un paramètre de filtres dans les requêtes

Lorsque j'appelle un endpoint

Je dis qu'il n'y a pas d'endpoint à appeler. Ce qui est ajouté ici est présent dans la définition des requêtes, end of story, jamais reçu jamais recevable dans l'URL.

## #11 - 12 avril 2021 18:17 - Nicolas Roche

- Fichier 0003-opendatasoft-add-facet-filters-50212.patch ajouté

- Fichier 0002-opendatasoft-use-APIError-into-endpoint-50212.patch ajouté

- Fichier 0001-opendatasoft-correct-tests-50212.patch ajouté

- Statut changé de Information nécessaire à Solution proposée

(compris, merci)

## #12 - 13 avril 2021 10:23 - Frédéric Péters

Oui c'est ça, modulo tu as encore un `**kwargs` qui va permettre n'importe quoi, je ne le pense pas nécessaire ici :

```
+     return self.resource.call_search(  
+         dataset=self.dataset,  
+         text_template=self.text_template,  
+         filter_expression=self.filter_expression,  
+         **kwargs,
```

## #13 - 13 avril 2021 12:22 - Nicolas Roche

- Fichier 0003-opendatasoft-add-facet-filters-50212.patch ajouté

J'ai explicité l'appel pour supprimer le kwargs.

## #14 - 27 avril 2021 18:26 - Frédéric Péters

J'ai vraiment du mal à suivre les versions qui tournent sans arriver à la bonne forme.

Bref sur la dernière je ne comprends pas pourquoi on a toujours :

```
+ def call_search(  
+     self, dataset=None, text_template='', filter_expression='', id=None, q=None, limit=None, **kwargs  
+ ):
```

alors que la fonction n'a pas à recevoir de paramètres inconnus.

Aussi, ce changement :

```
- def q(self, request, **kwargs):  
-     return self.resource.search(request, dataset=self.dataset, text_template=self.text_template, **kwargs  
)  
+ def q(self, **kwargs):  
+     return self.resource.call_search(  
+         self.dataset,  
+         self.text_template,  
+         self.filter_expression,  
+         kwargs.get('id'),  
+         kwargs.get('q'),  
+         kwargs.get('limit'),  
+     )
```

m'interloque totalement, pourquoi n'a-t-il pas pu être laissé comme il était ?

~

Si ça vient d'instructions mal formulées dans des commentaires précédents, désolé, sans doute je ne relis pas en me remettant totalement dans le code à chaque fois.

Je peux comprendre l'enchaînement :

- l'endpoint "q" continue à recevoir \*\*kwargs et donc potentiellement n'importe quoi
- celui-ci appelle la méthode "q" avec \*\*kwargs et donc potentiellement n'importe quoi
- à ce niveau on décide d'arrêter le n'importe quoi et passer uniquement les paramètres triés sur le volet
- à la méthode call\_search, sauf que celle-ci est modifiée dans le même temps pour accepter n'importe quoi

Ma perspective, c'est qu'on doit contrôler ce qui arrive sur les endpoints (i.e. les méthodes avec @endpoint), c'est-à-dire y avoir de vrais paramètres, ce qui en permet la description; qu'on peut y avoir \*\*kwargs pour juste ignorer le potentiellement n'importe quoi, mais que de ces endpoints, le \*\*kwargs ne doit pas sortir.

Voilà, en fait je ne relis pas toujours avec tout le contexte autour parce que ça me fait trop facilement réinterroger des chose qui sortent vraiment du cadre du patch, j'essaie d'éviter ça ici.

#### #15 - 28 avril 2021 16:37 - Nicolas Roche

- Fichier 0001-opendatasoft-add-facet-filters-50212.patch ajouté

Oui d'accord avec tout (j'ai raté un truc et c'est ce qui a soulevé les 2 interrogations ci-dessus).

Soit je retire le kwargs des paramètres de call\_search (pour qu'à ce niveau on décide d'arrêter le n'importe quoi et passer uniquement les paramètres triés sur le volet),  
soit je le laisse pour que la méthode "q" appelle call\_search avec le \*\*kwargs et donc potentiellement n'importe quoi.

J'étais parti sur la première version (mais j'avais oublié de retirer le kwargs des paramètres de call\_search). A te lire je suppose que tu préfères la seconde version qui génère un plus petit patch.  
Mes excuses ici pour le mauvais patch ci-dessus qui a tout embrouillé.

Rien à voir, mais je note ici qu'il me faudra ensuite mettre à jour la [page de wiki](#) :

Puis à créer une requête pour renseigner les autres paramètres :

```
...  
* filtre : les filtres 'refine' et 'exclude' à appliquer (https://help.opendatasoft.com/apis/ods-search-v1/#dataset-search-api)  
concaténés avec des '&'.  
ex: refine.continent_fr=Europe et Amérique du nord&exclude.category=Cultural&refine.country_fr=France
```

#### #16 - 28 avril 2021 16:55 - Frédéric Péters

J'étais parti sur la première version (mais j'avais oublié de retirer le kwargs des paramètres de call\_search).

Oui.

A te lire je suppose que tu préfères la seconde version qui génère un plus petit patch.

Non. Ça m'intéresse de savoir ce qui a laissé penser ça.

Dans ce patch tu introduis `call_search` qui est une nouvelle méthode qui n'a aucune raison d'avoir `**kwargs`.

Si ce n'est que dans Query/q ça continue à passer n'importe quoi qui est arrivé par l'URL, via l'endpoint "q" (qui m'a soulevé hier mille questions qui m'ont fait abandonné l'affaire), et le truc que je pensais clair c'était qu'on peut avoir un `**kwargs` sur un endpoint pour ne pas planter sur un paramètre qui serait inconnu, mais qu'on ne doit pas propager ça. (et visiblement pas clair).

#### #17 - 28 avril 2021 18:02 - Nicolas Roche

- Fichier `0001-opendatasoft-add-facet-filters-50212.patch` ajouté

A te lire je suppose que tu préfères la seconde version qui génère un plus petit patch.

(c'est ce que j'ai pensé en lisant : "m'interloque totalement, pourquoi n'a-t-il pas pu être laissé comme il était ?")

Dans ce patch tu introduis `call_search` qui est une nouvelle méthode qui n'a aucune raison d'avoir `**kwargs`.

oui j'ai complètement bugé, désolé.

Et donc patch avec Query/q qui ne passer plus n'importe quoi qui est arrivé par l'URL, via l'endpoint "q".

#### #21 - 25 juin 2021 14:32 - Frédéric Péters

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

J'étais presque à valider et dire que les développements supplémentaires imaginés, comme permettre un gabarit dans `filter_expression`, pour par exemple y avoir `{{now|date:"Y"}}` pour filtrer sur l'année actuelle, ce serait pour d'autres tickets.

Mais quand même je pense que c'est peu lisible de taper `refine.source=Ville et Eurométropole de Strasbourg&exclude.numero=42&exclude.numero=43`. et qu'on pourrait accepter les retours à la ligne, pour comme ça avoir une expression par ligne, et pouvoir écrire :

```
refine.source=Ville et Eurométropole de Strasbourg
exclude.numero=42
exclude.numero=43
```

Il s'agirait de modifier `help_text` pour dire qu'une expression par ligne (plutôt que séparées par des &) et quand il est passé `filter_expression=self.filter_expression`, plutôt faire un `'&'.join(self.filter_expression.splitlines())`.

#### #22 - 25 juin 2021 14:44 - Frédéric Péters

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

#### #23 - 25 juin 2021 17:42 - Nicolas Roche

- Fichier `0003-opendatasoft-add-facet-filters-50212.patch` ajouté

- Fichier `0002-opendatasoft-use-APIError-into-endpoint-50212.patch` ajouté

- Fichier `0001-opendatasoft-correct-tests-50212.patch` ajouté

une expression par ligne

oui, en effet.

#### #24 - 25 juin 2021 19:33 - Nicolas Roche

- Fichier `0003-opendatasoft-add-facet-filters-50212.patch` ajouté

(j'ai oublié de mettre à jour la migration)

## #25 - 27 août 2021 13:56 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

## #26 - 27 août 2021 14:19 - Nicolas Roche

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

```
commit cac06da89eela059bedd1b1ff2a50af0970256ac
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Fri Apr 9 11:39:17 2021 +0200
```

```
opendatasoft: add facet filters (#50212)
```

```
commit ecd314d714ba20fb2fc0adb5202b78971f759605
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Mon Apr 12 17:55:39 2021 +0200
```

```
opendatasoft: use APIError into endpoint (#50212)
```

```
commit 9b88f614aa8a6aa30afbb4e389f480c44532b999
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Mon Apr 12 17:54:29 2021 +0200
```

```
opendatasoft: correct tests (#50212)
```

## #27 - 30 août 2021 12:17 - Frédéric Péters

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

### Fichiers

0001-opendatasoft-add-facet-filters-50212.patch	9,27 ko	09 avril 2021	Nicolas Roche
0001-opendatasoft-add-facet-filters-50212.patch	8,4 ko	12 avril 2021	Nicolas Roche
0003-opendatasoft-add-facet-filters-50212.patch	9,64 ko	12 avril 2021	Nicolas Roche
0002-opendatasoft-use-APIError-into-endpoint-50212.patch	1,94 ko	12 avril 2021	Nicolas Roche
0001-opendatasoft-correct-tests-50212.patch	2,11 ko	12 avril 2021	Nicolas Roche
0003-opendatasoft-add-facet-filters-50212.patch	9,7 ko	13 avril 2021	Nicolas Roche
0001-opendatasoft-add-facet-filters-50212.patch	9,64 ko	28 avril 2021	Nicolas Roche
0001-opendatasoft-add-facet-filters-50212.patch	9,7 ko	28 avril 2021	Nicolas Roche
0002-opendatasoft-use-APIError-into-endpoint-50212.patch	1,91 ko	25 juin 2021	Nicolas Roche
0003-opendatasoft-add-facet-filters-50212.patch	9,81 ko	25 juin 2021	Nicolas Roche
0001-opendatasoft-correct-tests-50212.patch	2,11 ko	25 juin 2021	Nicolas Roche
0003-opendatasoft-add-facet-filters-50212.patch	9,81 ko	25 juin 2021	Nicolas Roche