

Passerelle - Development #29889

jsondatastore : endpoint de recherche

18 janvier 2019 14:49 - Emmanuel Cazenave

Statut:	Fermé	Début:	18 janvier 2019
Priorité:	Normal	Echéance:	
Assigné à:	Nicolas Roche	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Cas d'usage : une instance de connecteur jsondatastore utilisé comme référentiel d'association. Ce référentiel se constitue grâce à une démarche de déclaration d'association, un usager peut déclarer X associations. Dans le portail d'agent, une cellule de recherche, on aimerait qu'en y tapant un nom d'asso, ressorte l'utilisateur qui l'a déclaré. Et donc un endpoint qui rechercherait dans JsonData.text et renverrait une réponse exploitable par une cellule de recherche.			
Demandes liées:			
Lié à Passerelle - Development #37849: jsondatastore: ajouter &q=plop, filtre...		Fermé	21 novembre 2019

Historique

#1 - 13 novembre 2019 10:38 - Nicolas Roche

Est-ce que par chance [#35780](#) ne ferait-il pas l'affaire ?

#2 - 13 novembre 2019 13:17 - Benjamin Dauvergne

[#35780](#) ne renvoie qu'une ligne, la première qui matche, il faudrait plutôt gérer un paramètre q sur l'endpoint list je pense.

#3 - 13 novembre 2019 13:52 - Nicolas Roche

Non non, c'était justement l'objet de la demande de Paul dans ce ticket :

Ça m'arrangerait bien que cet appel renvoie une liste d'objets comme un simple GET sur /jsondatastore/whatever/data/.

et dans les tests :

```
resp = app.get('/jsondatastore/foobar/data/',
               params={'key1': 'val1', 'key2': 'val2'})
assert sorted([d['id'] for d in resp.json['data']]) == sorted([uuid2, uuid3])
```

#4 - 13 novembre 2019 15:30 - Benjamin Dauvergne

Ok reste donc seulement à implémenter des opérateurs de match différents voir simplement le faire pour la clé text via q, parce que pour l'instant ça ne permet qu'un match exact alors qu'il faudrait pour de la recherche :

- pouvoir rechercher une sous chaîne,
- que ce soit insensible aux accents et à la casse.

#5 - 20 novembre 2019 17:13 - Nicolas Roche

- Assigné à mis à Nicolas Roche

#6 - 20 novembre 2019 18:51 - Nicolas Roche

- Fichier 0001-jsondatastore-add-a-search-endpoint-29889.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Solution un peu moche parce que les nom des champs à renvoyer peuvent interférer avec les couples clé/valeurs passés pour la recherche.
ie: les chaînes 'url_field', 'text_field' et 'description_field' sont exclues des clés utilisables pour la recherche.

Peut-être qu'il faudrait des paramètres plus explicites : je pensais à 'projected_XXX_field' mais ça ne colle pas avec la définition d'une projection.

Pour tester :

Une instance de connecteur jsondatastore utilisée comme référentiel d'association. Ce référentiel se constitue grâce à une démarche de déclaration d'association, un usager peut déclarer X associations.

```
$ curl -X POST -d '{"text": "miam", "description": "miam miam", "url": "http://www.miam-asso.fr/", "usager": "manu"}' https://passerelle.dev.publik.love/jsondatastore/associations/data/create
$ curl -X POST -d '{"text": "miam", "description": "miam miam", "url": "http://www.miam-asso.fr/", "usager": "nico"}' https://passerelle.dev.publik.love/jsondatastore/associations/data/create
```

Dans le portail d'agent, une cellule de recherche, on aimerait qu'en y tapant un nom d'asso, ressorte l'usager qui l'a déclaré.

~/config/publik/settings/combo/settings.d/assos.py :

```
COMBO_SEARCH_SERVICES = {
    'user': {
        'label': "usager qui a declare l'association",
        'url': 'https://passerelle.dev.publik.love/jsondatastore/associations/search/?text_field=usager&description_field=usager&text=%(q)s',
    },
}
```

Et donc un endpoint qui rechercherait dans JsonData.text et renverrait une réponse exploitable par une cellule de recherche.

```
$ curl 'https://passerelle.dev.publik.love/jsondatastore/associations/search/?text=ia&text_field=usager' | json_pp
{
  "data" : [
    {
      "description" : "miam miam",
      "text" : "nico",
      "url" : "http://www.miam-asso.fr/"
    },
    {
      "description" : "miam miam",
      "text" : "manu",
      "url" : "http://www.miam-asso.fr/"
    }
  ],
  "err" : 0
}
```

#7 - 20 novembre 2019 19:09 - Frédéric Péters

Solution un peu moche parce que les nom des champs à renvoyer peuvent interférer avec les couples clé/valeurs passés pour la recherche.

Je pense que les résultats doivent être identiques à ce qu'on obtient via le endpoint list, pas inventer ici un système créant autre chose. Et exprimé ainsi ça devient évident qu'il s'agit "juste" d'accepter des paramètres supplémentaires sur cet endpoint, et je serais pour juste deux trucs, &whatevey=plop, filtre exact sur une clé, &id=plop, filtre exact sur l'attribut id, et &q=plop, filtre sur l'attribut text (mais pas exact, ici .lower()) et se contenter d'une sous-chaine).

#8 - 21 novembre 2019 11:56 - Nicolas Roche

- Lié à Development #37849: jsondatastore: ajouter &q=plop, filtre sur l'attribut text au endpoint list ajouté

#9 - 21 novembre 2019 11:57 - Nicolas Roche

- Statut changé de Solution proposée à En cours

#10 - 21 novembre 2019 18:15 - Nicolas Roche

- Fichier 0001-jsondatastore-add-a-search-endpoint-29889.patch ajouté

- Statut changé de En cours à Solution proposée

les résultats doivent être identiques à ce qu'on obtient via le endpoint list

Voici un path qui factorise le code du endpoint list (lui même adapté dans [#37849](#)).

#11 - 29 novembre 2019 12:32 - Thomas Noël

- Statut changé de Solution proposée à En cours

Comme disait Frédéric, il ne s'agit pas de créer un nouvel endpoint. Il faut rester dans "list", juste lui ajouter les paramètres optionnels.

#12 - 29 novembre 2019 18:44 - Nicolas Roche

Je pense que les résultats doivent être identiques à ce qu'on obtient via le endpoint list

en fait non, la structure retournée est différente :

```
## contenu du jsondatastore brut
{"data": [{
  "content": {
    "asso": "cyclofficine",
    "declarant": "nico"
  },
  "text": "cyclofficine",
  "creation_datetime": "2019-11-20T13:53:42.618Z",
  "id": "83f9a3a7705943528cf09374489ce651",
  "last_update_datetime": "2019-11-20T13:53:42.618Z"
}],
----
## contenu pour une cellule de recherche
{"data": [{
  "description": "cyclofficine",
  "text": "nico",
  "url": "http://quelconque.fr/"
}],
```

Surtout, je ne saurais pas quoi faire des 3 paramètres de projection dans le endpoint liste.

```
url_field='url', text_field='text', description_field='description'
```

#13 - 29 novembre 2019 19:28 - Thomas Noël

On voit qu'il faut juste ajouter des clés (url et description) dans les dicos des résultats de list. Laisser tout le reste, c'est toujours bien de l'avoir, et la cellule de recherche de Combo saura éventuellement l'exploiter.

En gros tu peux simplement modifier to_json et y ajouter deux clés:

```
'url': self.content.get('url') if self.content else None,
'description': self.content.get('description') if self.content else None,
```

Les truc_field sont inutiles.

(mais je relis ton patch et je constate qu'il ne s'applique pas sur master, ou "list" n'a pas de paramètre q... j'ai dû rater un bout.)

#14 - 30 novembre 2019 09:22 - Nicolas Roche

J'ai déporté l'ajout du paramètre q dans [#37849](#).

Merci pour les explications !

... mais j'aurais quand même besoin d'au moins un champs 'truc' pour savoir quel champs retourner :

une cellule de recherche, on aimerait qu'en y tapant un nom d'asso, ressorte l'utilisateur qui l'a déclaré.

il s'agirait de retourner l'utilisateur qui a déclaré l'association et qui donc serait un champs 'annexe'.

En gros j'ai compris :

```
select (déclarant) from Asso where description="cyclofficine"
```

Sauf peut-être si la cellule de recherche de Combo saura fouiller le JSON pour le retrouver ? Dis-moi.

#15 - 30 novembre 2019 12:46 - Frédéric Péters

Sauf peut-être si la cellule de recherche de Combo saura fouiller le JSON pour le retrouver ? Dis-moi.

Dans la définition d'un service de recherche dans Combo (COMBO_SEARCH_SERVICES), on peut définir les gabarits des libellés, descriptions et URL, ex:

```
COMBO_SEARCH_SERVICES = {
  'association': {
    'label': 'Search an association',
    'url': 'https://passerelle.whatever/.../?q=%(q)s',
    'hit_url_template': '{{content.url}}',
    'hit_label_template': '{{text}}',
  },
}
```

La seule exigence concernant la structure est {"data": [...]}, pas besoin d'adaptations au contenu.

#16 - 02 décembre 2019 17:10 - Nicolas Roche

Merci, j'ai fini par atterrir : les projections sont déjà gérées au travers des champs hit_XXX_template utilisable dans la configuration des cellules de recherche, lors du rendu de la cellule.

```
{
  "COMBO_SEARCH_SERVICES": {
    "association": {
      "label": "Search an association",
      "url": "https://passerelle.dev.publik.love/jsondatastore/associations/data/?q=%(q)s",
      "hit_url_template": "ici/{{content.usager}}",
      "hit_label_template": "{{text}}",
      "hit_description_template": "déclaré par {{content.usager}}"
    }
  }
}
```

En utilisant cette configuration, je constate qu'il n'y a besoin de rien de plus que ce qui est proposé par [#37849](#).
J'attends qu'il soit résolu pour fermer ici.

#17 - 05 décembre 2019 18:16 - Nicolas Roche

- Statut changé de En cours à Fermé

Fichiers

0001-jsondatastore-add-a-search-endpoint-29889.patch	4,18 ko	20 novembre 2019	Nicolas Roche
0001-jsondatastore-add-a-search-endpoint-29889.patch	4,14 ko	21 novembre 2019	Nicolas Roche