

Passerelle - Development #11057

csv datasource : supporter les operateurs <eq> <ne> <lt> <le> <gt> <ge>

26 mai 2016 11:29 - Josué Kouka

Statut:	Fermé	Début:	26 mai 2016
Priorité:	Normal	Echéance:	
Assigné à:	Josué Kouka	% réalisé:	100%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Supporter ces différents opérateurs, permettrait de faire des requetes tels que : data?street_name=charlemagne&street_side=l&street_start_number__lt=45&street_end_number__gt=45			
Demandes liées:			
Lié à Passerelle - Development #11023: connecteur csv (Nancy)		Fermé	23 mai 2016

Révisions associées

Révision 8f536fee - 13 juin 2016 16:26 - Josué Kouka

csv data source : add advanced filters support (#11057)

Historique

#1 - 26 mai 2016 11:29 - Josué Kouka

- Lié à Development #11023: connecteur csv (Nancy) ajouté

#2 - 26 mai 2016 15:50 - Josué Kouka

- Statut changé de Nouveau à En cours

#3 - 30 mai 2016 16:29 - Josué Kouka

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

#4 - 30 mai 2016 17:18 - Josué Kouka

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

ajout de csvoperators.py

#5 - 30 mai 2016 19:19 - Josué Kouka

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

- Patch proposed changé de Non à Oui

tous les tests passent

#6 - 30 mai 2016 19:41 - Frédéric Péters

Ça serait plus clair si l'opérateur utilisé pour gt était > et non < :

```
def gt_operator(i, v, case_insensitive=False):  
    return lambda x: int(v) < int(x[i])
```

Et peut-être ce serait plus clair si les deux opérandes étaient de même nature, et pas d'un côté l'index dans la ligne et de l'autre la valeur de comparaison ? La valeur tirée de la ligne étant alors tirée avant, dans le models.py.

~~~

Ajouter "icontains" ? Et "ieq" et "nieq" ? Comme ça il n'y aurait pas un "case\_insensitive" trimballé partout mais juste la compat :

```
if query and filters:  
    if case_insensitive:
```

```
filters = {'text__icontains': query}
else:
    filters = {'text__contains': query}
```

(à propos de ces lignes, dans le patch il y a un coup `if query and not filters` et puis un coup `if query and filters` qui fait proche la même chose)

~~~

Mais avant ça, plus généralement, je virerais `query` et `case_insensitive` de la signature du `get_data()`, je ferais plutôt que la vue crée les filtres adéquats.

~~~

`import csvoperators` : le mettre en tête de module, pas de raison à l'importer seulement là. (et plutôt `from . import csvoperators`).

~~~

Ce changement n'a pas de rapport, autre ticket autre commit :

```
- dialect = csv.Sniffer().sniff(content[:1024])
+ dialect = csv.Sniffer().sniff(content)
```

~~~

Vraiment un truc qu'il est utile de logger ? (on a déjà l'URL, qui nous permet de tout déduire, il me semble).

```
self.logger.info(filters)
```

~~~

Je reste vieux et peu à l'aise avec ces `yield` et `itertools` et générateurs, etc. Ça se ferait d'ajouter quelques commentaires, genre `"# get matches applying all filters on rows"` ?

~~~

Ça a déjà été bien pire mais il reste des endroits qui ne sont pas corrects par rapport à l'utilisation des espaces, genre `for key,value in filters.items():` (et c'est peut-être le seul, je n'ai pas fait d'inspection détaillée)

#### #7 - 02 juin 2016 15:47 - Josué Kouka

- Fichier `0004-csv-data-source-makes-view-generates-right-filters-1.patch` ajouté
- Fichier `0003-csv-data-source-add-post-method-support-for-tests-11.patch` ajouté
- Fichier `0002-csv-datasource-use-web-client-for-tests-11057.patch` ajouté
- Fichier `0001-csv-data-source-add-advanced-filters-support-11057.patch` ajouté

Ça serait plus clair si l'opérateur utilisé pour `gt` était `>` et non `<` :

```
> def gt_operator(i, v, case_insensitive=False):
> return lambda x: int(v) < int(x[i])
>
```

yep

Je reste vieux et peu à l'aise avec ces `yield` et `itertools` et générateurs, etc. Ça se ferait d'ajouter quelques commentaires, genre `"# get matches applying all filters on rows"` ?

done

- J'ai migré la génération des bons filtres dans la vue
- J'ai ajouté les tests pour la méthode POST

#### #8 - 02 juin 2016 16:54 - Josué Kouka

- Fichier `0001-csv-data-source-makes-view-generates-right-filters-1.patch` ajouté

mise à jour de la description des opérateurs

**#9 - 03 juin 2016 13:00 - Josué Kouka**

- Fichier 0005-csv-data-source-makes-view-generates-right-filters-1.patch ajouté
- Fichier 0004-csv-data-source-add-post-method-support-for-tests-11.patch ajouté
- Fichier 0003-csv-datasource-use-web-client-for-tests-11057.patch ajouté
- Fichier 0002-csv-data-source-add-advanced-filters-support-11057.patch ajouté
- Fichier 0001-csv-datasource-add-POST-method-support.patch ajouté

patch à jour

**#10 - 03 juin 2016 13:59 - Frédéric Péters**

0001-csv-datasource-add-POST-method-support.patch  
0004-csv-data-source-add-post-method-support-for-tests-11.patch

→ [#11059](#)

0002-csv-data-source-add-advanced-filters-support-11057.patch  
0005-csv-data-source-makes-view-generates-right-filters-1.patch

devraient être un seul patch, ça facilitera la relecture.

**#11 - 07 juin 2016 10:19 - Josué Kouka**

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

J'ai enlevé le support de POST et fait un seul patch sur les filtres avancés (logique + tests)

**#12 - 07 juin 2016 11:19 - Josué Kouka**

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

**#13 - 07 juin 2016 11:29 - Josué Kouka**

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

suppression scorie

**#14 - 07 juin 2016 12:01 - Benjamin Dauvergne**

Je pense que nie\_operator devrait être ine\_operator pour garder la logique de nommage (ne pourrait-on avoir igt, ilt, ige, ile au passage ?), à corriger aussi dans le template detail.

**#15 - 07 juin 2016 12:18 - Josué Kouka**

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

Benjamin Dauvergne a écrit :

Je pense que nie\_operator devrait être ine\_operator pour garder la logique de nommage (ne pourrait-on avoir igt, ilt, ige, ile au passage ?), à corriger aussi dans le template detail.

je ne trouve malheureusement pas de cas où l'on pourrait utiliser *igt*, *ilt*, *ige*, *ile*

**#16 - 07 juin 2016 13:53 - Benjamin Dauvergne**

Je le proposais juste par cohérence, la source CSV étant destinée à un usage assez large (entendre par des fonctionnels en autonomie) je pense qu'il faut viser du générique un minimum (et c'est 8 lignes de code).

**#17 - 07 juin 2016 14:42 - Josué Kouka**

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

**#18 - 07 juin 2016 14:47 - Benjamin Dauvergne**

Et donc discussion par Jabber:

```
josue mar. 07 juin 2016, 14:04:10
par rapport a https://dev.entrouvert.org/issues/11057. L'idée de faire du igt, ige, ile, ilt pour des str me d
érange un peu
enfin, sauf si l'on veut faire de la comparaison de valeurs ASCII
josue mar. 07 juin 2016, 14:09:49
s/str/char/
bdauvergne@im.libre-entreprise.com mar. 07 juin 2016, 14:10:50
ça de toute façon oui il faut gérer les règles de collation
on s'en sort pas sinon
josue mar. 07 juin 2016, 14:13:32
ok
bdauvergne@im.libre-entreprise.com mar. 07 juin 2016, 14:15:44
In [9]: locale.setlocale(locale.LC_ALL, 'fr_FR.UTF-8')
Out[9]: 'fr_FR.UTF-8'
```

```
In [10]: for x in sorted(l, cmp=locale.strcoll):
        print x
        ....:
à
ä
e
é
è
l = [u'è', u'à', u'è', u'e', u'à']
le truc chiant c'est le setlocale
Django a un traitement interne des locales et c'est une action globale
```

Mon avis c'est de ne pas gérer les règles de collation tout de suite mais dans les opérateurs de comparaison remplacer les < et > par une fonction `compare_string(left, right)` définie par `compare_string = cmp`, et donc on remplace `left < right` par `compare_string(left, right) < 0` et on pourra voir plus tard comment on veut traiter les locales (avec `locale.setlocale()` et `locale.strcoll()`, ou `pyluc` ou je ne sais quoi).

#### #19 - 07 juin 2016 15:12 - Josué Kouka

- Fichier `0001-csv-data-source-add-advanced-filters-support-11057.patch` ajouté

#### #20 - 07 juin 2016 18:16 - Thomas Noël

En dehors de la lecture du patch, mon avis, parce que je comprends pas l'idée des "lt/gt" pour les strings. Pour moi ça lt/gt n'ont de sens que pour les nombres, et pour des filtres sur un CSV, ça ne sera disponible que pour des nombres. En effet, on ne connaît pas le type des colonnes du CSV, et quand il y aura un filtre "foo\_lt=4", on doit considérer que les valeurs de foo sont entières (à la rigueur décimales si `foo_lt=4.1`). Voilà c'était mon avis du soir.

#### #21 - 07 juin 2016 19:06 - Josué Kouka

- Lié à Bug #11258: `csv datasource` : augmenter la taille du contenu passé au sniffer ajouté

#### #22 - 07 juin 2016 23:54 - Thomas Noël

- Lié à Bug #11258: `csv datasource` : augmenter la taille du contenu passé au sniffer supprimé

#### #23 - 08 juin 2016 16:06 - Josué Kouka

- Fichier `0001-csv-data-source-add-advanced-filters-support-11057.patch` ajouté

- ajout d'une classe `Lookup`
- modification du `try/catch` dans le générateur de filtres

#### #24 - 08 juin 2016 16:23 - Josué Kouka

- Fichier `0001-csv-data-source-add-advanced-filters-support-11057.patch` ajouté

suppression scores

#### #25 - 08 juin 2016 16:55 - Josué Kouka

- Fichier `0001-csv-data-source-add-advanced-filters-support-11057.patch` ajouté

#### #26 - 13 juin 2016 10:11 - Serghei Mihai

Si la valeur est 0, la fonction `is_int` va retourner `False` dans le contexte booléen et la comparaison en tant qu'entiers ne se fera pas. Je verrais plutôt:

```
def is_int(value):
    try:
        v = int(value)
        return True
    except (ValueError, TypeError):
        return False
```

#### #27 - 13 juin 2016 10:27 - Josué Kouka

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

merci Serghei

#### #28 - 13 juin 2016 10:53 - Frédéric Péters

modulo l'espace manquant,

```
except (ValueError, TypeError)
    ^
```

#### #29 - 13 juin 2016 10:56 - Frédéric Péters

Aussi,

```
def test_unknown_operator(client, setup):
    [...]
    assert result['err'] == 1
    assert result['err_class'] == 'AttributeError'
```

Ce serait bien de fournir une info plus précise.

#### #30 - 13 juin 2016 11:55 - Josué Kouka

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

#### #31 - 13 juin 2016 16:17 - Josué Kouka

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

fixe une indentation incohérente et des erreurs PEP8

#### #32 - 13 juin 2016 16:29 - Josué Kouka

- Fichier 0001-csv-data-source-add-advanced-filters-support-11057.patch ajouté

remplace '\_\_\_' par lookups.DELIMITER

#### #33 - 13 juin 2016 18:48 - Serghei Mihai

Go

#### #34 - 13 juin 2016 18:58 - Josué Kouka

- Statut changé de En cours à Résolu (à déployer)

- % réalisé changé de 0 à 100

#### #36 - 04 août 2018 12:31 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

### Fichiers

|                                                                 |         |              |             |
|-----------------------------------------------------------------|---------|--------------|-------------|
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 6,05 ko | 30 mai 2016  | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 7,48 ko | 30 mai 2016  | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 9,11 ko | 30 mai 2016  | Josué Kouka |
| 0003-csv-data-source-add-post-method-support-for-tests-11.patch | 8,87 ko | 02 juin 2016 | Josué Kouka |
| 0004-csv-data-source-makes-view-generates-right-filters-1.patch | 7,45 ko | 02 juin 2016 | Josué Kouka |
| 0002-csv-datasource-use-web-client-for-tests-11057.patch        | 9,35 ko | 02 juin 2016 | Josué Kouka |

|                                                                 |         |              |             |
|-----------------------------------------------------------------|---------|--------------|-------------|
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 9,3 ko  | 02 juin 2016 | Josué Kouka |
| 0001-csv-data-source-makes-view-generates-right-filters-1.patch | 8,93 ko | 02 juin 2016 | Josué Kouka |
| 0005-csv-data-source-makes-view-generates-right-filters-1.patch | 8,93 ko | 03 juin 2016 | Josué Kouka |
| 0004-csv-data-source-add-post-method-support-for-tests-11.patch | 8,87 ko | 03 juin 2016 | Josué Kouka |
| 0003-csv-datasource-use-web-client-for-tests-11057.patch        | 9,35 ko | 03 juin 2016 | Josué Kouka |
| 0002-csv-data-source-add-advanced-filters-support-11057.patch   | 9,3 ko  | 03 juin 2016 | Josué Kouka |
| 0001-csv-datasource-add-POST-method-support.patch               | 1,91 ko | 03 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 19,5 ko | 07 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 19,4 ko | 07 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 19,3 ko | 07 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 19,3 ko | 07 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 19,6 ko | 07 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 19,7 ko | 07 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 20 ko   | 08 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 20 ko   | 08 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 19,8 ko | 08 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 19,8 ko | 13 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 20,1 ko | 13 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 20,6 ko | 13 juin 2016 | Josué Kouka |
| 0001-csv-data-source-add-advanced-filters-support-11057.patch   | 20,6 ko | 13 juin 2016 | Josué Kouka |