

w.c.s. - Development #43328

accès général / requêtes sur les fiches

26 mai 2020 14:43 - Frédéric Péters

Statut:	Fermé	Début:	26 mai 2020
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Dans une démarche de réservation il y a une série de champs, quantité de chaises, quantité de tables, etc.

Il y a un total à calculer, $\text{form_var_nb_chaises} * 2 + \text{form_var_nb_tables} * 5$, etc.

Plutôt que maintenir ces tarifs dans le workflow de réservation, on pourrait vouloir ça dans des fiches, genre une fiche nom: Chaises, tarif: 2; ça pourrait aussi se retrouver exploité dans un champ commentaire dans la démarche, pour afficher le détail des tarifs, et ne pas demander ainsi à maintenir ceux-ci à différentes endroits.

Bref, de là, j'en suis venu à dire que ce qu'il faut, c'est un accès requêtage/générique, qui permettrait

```
{{cards.materiel.objects.filter_by_nom|filter_value:"Chaises"|first|get:"tarif"}}
```

(et c'est un peu moche, quand même).

Bref, draft à réfléchir.

Révisions associées

Révision 7087f4a0 - 16 juillet 2020 19:23 - Frédéric Péters

general: add filtering methods to lazy querysets (#43328)

Révision ba0f3a4b - 16 juillet 2020 20:16 - Frédéric Péters

general: add template access to cards and forms (#43328)

Révision 9ebb81bd - 16 juillet 2020 20:16 - Frédéric Péters

general: allow slicing querysets (#43328)

Révision 1e83b2b6 - 16 juillet 2020 20:16 - Frédéric Péters

general: add same_user queryset "filter" (#43328)

Historique

#1 - 26 mai 2020 14:44 - Frédéric Péters

- Fichier 0001-general-add-global-access-to-cards-data-43328.patch ajouté

- Patch proposed changé de Non à Oui

#2 - 26 mai 2020 14:50 - Pierre Cros

Perso ça me plairait beaucoup de pouvoir faire ça.

#3 - 26 mai 2020 21:26 - Benjamin Dauvergne

Si on se limite à ne requêter que par des champs de type "slug" (conforme à un identifiant python) ou entier, on pourrait se passer de filtres de template.

```
class QuerySet(object):
    def __init__(self, formdef, query):
        self.formdef = data_class
        self.query = query
```

```
def __iter__(self):
    return self.formdef.data_class().select(self.query, iterator=True)
```

```
def __getattr__(self, name):
    for field in self.formdef.fields:
        if field.varname == name:
            return FieldQueryset(self, field)
    raise AttributeError(name)
```

```
def first(self):
    for formdata in self:
        return formdata
    return None
```

```
class FieldQueryset(object):
```

```
    def __init__(self, qs, field):
        self.qs = qs
        self.field = field
```

```
    def __getattr__(self, name):
        return QuerySet(self.qs.formdef, And(self.query, Equal(field.varname, name)))
```

```
    def __iter__(self):
        for formdata in self.qs:
            yield formdata.data.get(self.field.id)
```

Avec ça on peut écrire directement `{{ cards.material.objects.nom.Chaise.first.tarif }}` mais aussi `{% for nom in cards.materiel.objects.nom %}{% nom %}{% endfor %}`.

Mais franchement je trouverai plus simple d'inventer un langage de requête et de faire :

```
{% object-query "SELECT ONE tarif FROM materiel WHERE name = 'Chaise'" as tarif %}
```

ou

```
{% object-query "SELECT * FROM materiel WHERE int(tarif) > 0" as materiels_payants %}
```

avec une grammaire PEG (via par exemple <https://github.com/erikrose/parsimonious>) ça irait assez vite de définir un tel langage et ce sera toujours plus joli à lire que des filtres.

#4 - 26 mai 2020 21:37 - Frédéric Péters

Non.

#5 - 26 mai 2020 21:37 - Frédéric Péters

- *Description mis à jour*

(j'ai corrigé la description)

#6 - 27 mai 2020 05:55 - Pierre Cros

Ce n'est pas plus simple pour un admin fonctionnel d'apprendre un nouveau langage de requête, c'est même vraiment à éviter au moment où on essaye d'homogénéiser en se débarrassant de Python. Encore une fois on ne s'adresse pas à des développeurs, la mécanique des filtres ils ont du l'apprendre (un peu), restons avec ça.

Peut-être que c'est pas très esthétique ou performant, mais en ce qui me concerne j'ai compris immédiatement ce qui était fait avec les filtres.

#7 - 27 mai 2020 11:16 - Benjamin Dauvergne

Je pensais que c'était un ticket pour discuter, très bien que ce ne soit pas le cas.

#8 - 28 mai 2020 08:01 - Pierre Cros

Quelqu'un pour relire le patch du coup ?

#9 - 28 mai 2020 09:24 - Frédéric Péters

- *Patch proposed changé de Oui à Non*

Malgré la case à cocher, c'est le statut qui compte, il n'y a pas de patch proposé (le fichier attaché ne contient pas de tests, il faudra passer par jenkins, etc.).

#11 - 25 juin 2020 17:03 - Lauréline Guérin

notes de l'EO day: pas de solution, pas d'idée pour le moment.
A rediscuter.

#12 - 25 juin 2020 18:56 - Frédéric Péters

(pour info je compte bien reprendre le patch attaché, le continuer avec des tests, et avancer peut-être jusqu'au nécessaire pour gérer le cas d'usage "pas de doublon").

#15 - 28 juin 2020 14:48 - Frédéric Péters

- Fichier 0002-general-add-template-access-to-cards-and-forms-43328.patch ajouté

- Fichier 0001-general-add-filtering-methods-to-lazy-queriesets-4332.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

L'intitulé du ticket l'annonçait déjà, deux sujets :

- accès général, c'est 0002, c'est ajouter au contexte global une source "forms" et une source "cards", comme il y avait déjà un accès aux webservices et aux sources de données;
- requêtes sur les fiches, c'est 0001, c'est en fait continuer LazyFormDefObjectsManager en lui ajoutant quelques méthodes.

#16 - 29 juin 2020 21:02 - Frédéric Péters

- Fichier 0004-general-add-same_user-queryset-filter-43328.patch ajouté

- Fichier 0003-general-allow-slicing-queriesets-43328.patch ajouté

Dans la branche désormais également,

- 0003, pour autoriser les slices sur les queryset, (tombé dessus en faisant le test pour le 0004),
- 0004, pour ajouter un filtre "same_user" (qui est un truc fait dans le script externe de #44561).

#17 - 16 juillet 2020 17:03 - Thomas Noël

- Fichier 0005-general-allow-lazy-value-in-filter-filters-43328.patch ajouté

0001 : il manque un peu de "unlazy" dans les filtres, ça sera nécessaire dans la gestion anti-doublon quand il faudra faire du « ...|filter_value:form_var_siren ». Ci-joint une proposition pour gérer ça, à merger ou pas avec 0001.

Aussi dans le test on voit que `{{form_objects|filter_by:"foo_foo"|filter_value:"bar"|count}}` renvoie y compris les objets avec le statut "draft"... Allez, disons que le filtrage par statut fera l'objet d'un futur ticket/patch.

0002 : ok mais...

Viendra vite le besoin d'un templatetags tel que « `cards|objects:form_option_foo` » pour pouvoir récupérer une queryset sur les fiches du slug nommé par `form_option_foo`. Et alors on préférera documenter la syntaxe générale `{{ cards|objects:"foo"|... }}` au lieu de `{{ cards.foo.objects|... }}`

Peut-être que ça pourrait faire partie de ce 0002 ?

0003 : ok

0004 : ok mais... on imaginerait donc ici un `{{ cards.foo.objects.same_user|... }}` ? Là encore je serai pour une version filtre `{{ cards|objects:"foo"|same_user|... }}`

#18 - 16 juillet 2020 19:40 - Frédéric Péters

Aussi dans le test on voit que `{{form_objects|filter_by:"foo_foo"|filter_value:"bar"|count}}` renvoie y compris les objets avec le statut "draft"... Allez, disons que le filtrage par statut fera l'objet d'un futur ticket/patch.

Non les brouillons sont bien exclus; on est à 7 parce que en plus du for i in range(6): il y a la fixture variable_test_data.

Branche poussée avec les modifications suggérées.

#19 - 17 juillet 2020 00:24 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Frédéric Péters a écrit :

Aussi dans le test on voit que `{{form_objects|filter_by:"foo_foo"|filter_value:"bar"|count}}` renvoie y compris les objets avec le statut "draft"...
Allez, disons que le filtrage par statut fera l'objet d'un futur ticket/patch.

Non les brouillons sont bien exclus; on est à 7 parce que en plus du for i in range(6): il y a la fixture variable_test_data.

J'avais raté cette fixture, désolé.

Branche poussée avec les modifications suggérées.

Et tout me paraît plutôt assez chouette ainsi.

#20 - 17 juillet 2020 07:12 - Frédéric Péters

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

```
commit 1e83b2b6293615ad9f1198f092f89a79870409b1
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Mon Jun 29 16:49:01 2020 +0200
```

```
general: add same_user queryset "filter" (#43328)
```

```
commit 9ebb81bd093baa8e5f1014a5d4f2ce7f32ce1670
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Mon Jun 29 16:48:38 2020 +0200
```

```
general: allow slicing querysets (#43328)
```

```
commit ba0f3a4bb037c2c11ccfc7caade3a7268b4e463e
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Sun Jun 28 09:07:05 2020 +0200
```

```
general: add template access to cards and forms (#43328)
```

```
commit 7087f4a0610cbfe5d5f75846396d889986f4e773
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Sun Jun 28 08:51:38 2020 +0200
```

```
general: add filtering methods to lazy querysets (#43328)
```

#21 - 17 juillet 2020 09:16 - Frédéric Péters

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

Fichiers

0001-general-add-global-access-to-cards-data-43328.patch	3,9 ko	26 mai 2020	Frédéric Péters
0002-general-add-template-access-to-cards-and-forms-43328.patch	4,95 ko	28 juin 2020	Frédéric Péters
0001-general-add-filtering-methods-to-lazy-querysets-4332.patch	5,71 ko	28 juin 2020	Frédéric Péters
0004-general-add-same_user-queryset-filter-43328.patch	2,53 ko	29 juin 2020	Frédéric Péters
0003-general-allow-slicing-querysets-43328.patch	854 octets	29 juin 2020	Frédéric Péters
0005-general-allow-lazy-value-in-filter-filters-43328.patch	1,8 ko	16 juillet 2020	Thomas Noël