

w.c.s. - Development #27709

Filtres Django pour les opérations mathématiques

31 octobre 2018 16:12 - Pierre Cros

Statut:	Fermé	Début:	31 octobre 2018
Priorité:	Normal	Echéance:	
Assigné à:	Nicolas Roche	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Je le mets pas dans la doc encore parce que ça me semble plus compliqué que le reste.			
Mais en plus du add existant, faudrait un subtract, un multiple et un divide (et c'est moche de toute façon).			
Demandes liées:			
Lié à Combo - Development #41868: Ajouter la conversion décimal sur les filtre...		Fermé	20 avril 2020

Révisions associées

Révision 69f4fe62 - 04 mars 2019 13:49 - Nicolas Roche

templatetags: add mathematics filters (#27709)

Historique

#1 - 31 octobre 2018 16:51 - Frédéric Péters

Pour information quand même, sur l'ensemble des conditions (page pré et post, champs, workflows) des wcs de prod, je trouve :

- 0 addition
- 0 soustraction
- 0 multiplication
- 0 division

#2 - 31 octobre 2018 17:10 - Thomas Noël

D'une discussion jabber avec Pierre, on parle ici plutôt de calculs, par exemple de coût à envoyer dans un lingo.

#3 - 31 octobre 2018 17:53 - Pierre Cros

Yes les calculs dans les données de traitement par exemple.

#4 - 29 janvier 2019 17:57 - Thomas Noël

- Assigné à mis à Thomas Noël

#5 - 18 février 2019 14:51 - Nicolas Roche

- Assigné à changé de Thomas Noël à Nicolas Roche

#6 - 19 février 2019 09:17 - Nicolas Roche

- Statut changé de Nouveau à En cours

#7 - 19 février 2019 14:47 - Nicolas Roche

- Fichier 0001-add-mathematics-filters-27709.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Les opérateurs font un peu de magie pour aider les utilisateurs :

- add et subtract : si un opérateur est nul alors on le traite comme un 0 (le résultat est donné par l'autre opérateur)
- multiply et divide : si un opérateur est nul alors le résultat est la chaîne vide (on aurait pu multiplier ou diviser par 1)

Peut-être que ce n'est pas le comportement souhaitable... dites-moi.

#8 - 19 février 2019 14:59 - Frédéric Péters

Attention dans le patch `subtract` plutôt que `subStract`.

#9 - 19 février 2019 15:07 - Thomas Noël

Sur la forme :

- dans le test, y'a un `"import qommon.errors"` en trop au début
- on s'autorise des lignes de 100 caractères de large, donc les `with pytest.raises(Exception, match=r'.*Could not find variable at start.*')` doivent pouvoir tenir sur une ligne, ça sera plus joli
- j'ai tendance à mettre les cas non-passants (ces `with pytest.raises`) en fin du test... Globalement j'organiserais les tests ainsi, idem dans les sections plus bas, par exemple sur `divide` ou tu testes d'abord ce qui ne marche pas. Même si c'est pas toujours vrai, les tests sont également un peu une façon de documenter un truc, donc autant commencer par documenter "comment ça marche" avant de montrer comment ça plante.
- comme on dit en Python, "explicit is better than implicit" : `return v1 if v1 is not None else "` c'est pas très lisible, ajoute plutôt des `"if term1 is None: term1 = 0"` après le parsing. On comprendra qu'il s'agit de valeur par défaut en cas de crash du parsing.
- on ne nomme pas des variables "v1" ou "v2" en Python, c'est trop court... utilise plutôt `term1` et `term2`

Aussi : les tests sur `|decimal` sont inutiles ici, déjà fait ailleurs (dans `test_templates.py`). Tu peux les virer, plus un patch est petit mieux c'est :)

Enfin, dans les tests il faudra ajouter des calculs avec l'usage de variables qui viennent du formulaire (`formdata`). On peut voir ça ensemble tout à l'heure.

#10 - 21 février 2019 17:14 - Nicolas Roche

- Fichier `0001-templatetags-add-mathematics-filters-27709.patch` ajouté

dites-moi si j'ai zappé quelque-chose...

sinon c'est assez redondant et peut-être que vous auriez quelque chose à me suggérer pour alléger le code des filtres.

#11 - 22 février 2019 12:40 - Thomas Noël

Il faut essayer d'être plus systématique dans tes tests, car sinon tu ne va pas bien couvrir le code (va sur le lien du jenkins qui s'affiche ici, clique sur "Cobertura Coverage Report", puis sur `"qommon.templatetags"` puis `"qommon.py"` : tu verras que tes tests ne sont pas passés par le `"ZeroDivisionError"`)

Pour chaque opérateur, je ferais, en commençant toujours par "ce qui doit fonctionner" :

- test avec deux décimaux
- test avec un nombre sous forme de chaîne à gauche
- test avec un nombre sous forme de chaîne à droite
- test avec deux nombres sous forme de chaîne
- test avec une string-qui-n-est-pas-un-nombre à gauche
- test avec une string-qui-n-est-pas-un-nombre à droite
- test avec deux string-qui-ne-sont-pas-un-nombre

- ... et la division par zéro.

Pour réduire le code :

- gère la dé-lazyfication dans `parse_decimal`
- ajoute une valeur par défaut de retour sur `parse_decimal(value, default=None)`
... comme ça tu pourras faire `< value = parse_decimal(value, Decimal(0)) >` pour l'addition, par exemple.

Aussi, plutôt que `value` et `arg`, j'aurai eu tendance à utiliser `"term1"` et `"term2"`, mais bon, c'est totalement perso et ton choix de `value, arg` est bien aussi (c'est la notation classique dans un `templatetags`, pourrait-on dire)

#12 - 22 février 2019 17:32 - Nicolas Roche

- Fichier `0001-templatetags-add-mathematics-filters-27709.patch` ajouté

En effet, la division par zéro plantait.

Idem, dites-moi s'il vous voyez des choses qui ne sont pas d'équerre.

Pour la "dé-lazyfication" je vais le faire dans une autre ticket ([#30852](#)) car ça impacte également l'opérateur `"decimal"`.

#13 - 22 février 2019 17:46 - Thomas Noël

Pour suivre la logique `add` et `subtract` qui essayent de renvoyer un `decimal` même quand un des termes est foireux, sur `multiply` je renverrai bien `Decimal(0)` quand l'un des deux termes est `None`.

Et pour `divide` :

- renvoyer `"` si `term2` est `None` (vu comme une division par zéro)

- renvoyer Decimal(0) si term1 est None

Truc que je ne vois que maintenant : ton test "test_mathematics_filters(pub, variable_test_data):" ne fait en fait appel à aucune variable de formulaire. Tu peux en supprimer les 6 premières lignes... mais surtout il devrait plutôt être dans tests/test_templates.py, à la fin, après test_decimal_templatetag

#14 - 25 février 2019 14:17 - Nicolas Roche

- Fichier 0001-templatetags-add-mathematics-filters-27709.patch ajouté

Idem, dites-moi si ça passe comme ça...

#15 - 25 février 2019 14:21 - Nicolas Roche

- Fichier 0001-templatetags-add-mathematics-filters-27709.patch ajouté

oups, petit oubli, désolé.

#16 - 25 février 2019 14:27 - Thomas Noël

J'ajouterai des tests qui montrent que le résultat est bien un decimal, parce que c'est ce qui sera attendu et utilisé dans les conditions, pour faire des comparaisons. Et donc au niveau de test_lazy_formdata_mathematics_filters, ajouter des lignes du genre :

```
condition = Condition({'type': 'django', 'value': 'form_var_term1|add:form_var_term2 == 7'})
assert condition.evaluate() is True
condition = Condition({'type': 'django', 'value': 'form_var_term1|add:form_var_term2 == 7.0'})
assert condition.evaluate() is True
condition = Condition({'type': 'django', 'value': 'form_var_term1|add:form_var_term2 == "7"|decimal'})
assert condition.evaluate() is True
condition = Condition({'type': 'django', 'value': 'form_var_term1|add:form_var_term2 > 6'})
assert condition.evaluate() is True
condition = Condition({'type': 'django', 'value': 'form_var_term1|add:form_var_term2 > 8'})
assert condition.evaluate() is False
```

et le même genre de truc pour les autres opérateurs.

#17 - 25 février 2019 16:11 - Nicolas Roche

- Fichier 0001-templatetags-add-mathematics-filters-27709.patch ajouté

voilà, dites-moi...

#18 - 25 février 2019 17:03 - Thomas Noël

Sounds good, mais on va pas pousser ça cette semaine, ça sera pour la release prochaine (mi mars).

A noter un mini-détail :

```
    '9': '2018-07-31',
+    '10': '3',
+    '11': '4'
}
```

Il faut penser ici à l'avenir et pour que les prochains patches n'aient pas à modifier ta dernière ligne, termine-la par une virgule : '11': '4', (comme pour la ligne du '9': ...)

Tiens, truc auquel je pense maintenant : vérifier que {{form_var_foo_foo|decimal}} ne plante pas, et renvoie bien le nombre 0 (idem avec form_var_date, form_var_itemsfield et les autres). C'est toujours ça qu'on aura de moins comme trace au cas où :)

#19 - 25 février 2019 17:24 - Nicolas Roche

- Fichier 0001-templatetags-add-mathematics-filters-27709.patch ajouté

Questions :

- True|decimal renvoie '0', c'est bon ?
- pourquoi 'from wcs.conditions import Condition' dans test_templates.py ?

#20 - 25 février 2019 18:29 - Nicolas Roche

- Fichier 0001-templatetags-add-mathematics-filters-27709.patch ajouté

désolé, j'ai laissé trainé un 'assert False' (pour debug)

#21 - 26 février 2019 10:01 - Thomas Noël

Nicolas Roche a écrit :

Questions :

- True|decimal renvoie '0', c'est bon ?

Mouais... Bon, on va dire que oui, c'est comme ça :)

- pourquoi 'from wcs.conditions import Condition' dans test_templates.py ?

Un import qui traîne... Ca arrive, on en a par-ci par-là, on fait pas toujours assez attention (surtout sur les tests qui ne sont même pas pylintés). On peut les nettoyer de temps à autre, mais pas "en cachette" -- je veux dire, par exemple, ne pas nettoyer cet import dans ton patch qui n'a rien à voir. Plutôt faire un autre mini-patch genre «tests: remove unused import (trivial)», même sans ticket tellement c'est bateau.

#22 - 26 février 2019 10:02 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Je valide ton patch, mais attention à ne pas l'envoyer dans master : attendre lundi 4 mars (en gros "après la mise en prod de ce jeudi + 2 jours").

#23 - 04 mars 2019 13:56 - Nicolas Roche

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

```
commit 69f4fe62b067c38b87adbe636abcd066e1f9092a
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Mon Feb 25 10:58:55 2019 +0100
```

```
templatetags: add mathematics filters (#27709)
```

Avec le pad de thomas sur les filtres mis à jour : https://pad.libre-entreprise.org/p/publik_templatetags

#24 - 05 mars 2019 18:16 - Frédéric Péters

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

#25 - 20 avril 2020 10:00 - Nicolas Roche

- Lié à Development #41868: Ajouter la conversion décimal sur les filtres mathématiques par défaut ajouté

Fichiers

0001-add-mathematics-filters-27709.patch	6,51 ko	19 février 2019	Nicolas Roche
0001-templatetags-add-mathematics-filters-27709.patch	6,85 ko	21 février 2019	Nicolas Roche
0001-templatetags-add-mathematics-filters-27709.patch	9,6 ko	22 février 2019	Nicolas Roche
0001-templatetags-add-mathematics-filters-27709.patch	6,94 ko	25 février 2019	Nicolas Roche
0001-templatetags-add-mathematics-filters-27709.patch	6,61 ko	25 février 2019	Nicolas Roche
0001-templatetags-add-mathematics-filters-27709.patch	9,22 ko	25 février 2019	Nicolas Roche
0001-templatetags-add-mathematics-filters-27709.patch	10 ko	25 février 2019	Nicolas Roche
0001-templatetags-add-mathematics-filters-27709.patch	10 ko	25 février 2019	Nicolas Roche