

Combo - Development #41868

Ajouter la conversion décimal sur les filtres mathématiques par défaut

20 avril 2020 09:44 - Nicolas Roche

Statut:	Fermé	Début:	20 avril 2020
Priorité:	Normal	Echéance:	
Assigné à:	Nicolas Roche	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Actuellement les décimaux ne sont pas pris en compte sur : add, subtract, multiply, divide.			
Demandes liées:			
Lié à w.c.s. - Development #27709: Filtres Django pour les opérations mathéma...		Fermé	31 octobre 2018
Lié à Combo - Development #41699: Ajouter le filtre add		Fermé	14 avril 2020
Lié à Combo - Development #41698: Ajouter le filtre subtract		Fermé	14 avril 2020
Lié à Combo - Development #41697: Ajouter le filtre multiply		Fermé	14 avril 2020
Lié à Combo - Development #41700: Ajouter le filtre divide		Fermé	14 avril 2020
Lié à w.c.s. - Development #42062: permettre les manipulation de chaine avec ...		Fermé	23 avril 2020
Lié à Combo - Bug #42650: Erreur sur le filtre add		Fermé	08 mai 2020
Lié à w.c.s. - Development #42653: Simplification du filtre add		Rejeté	08 mai 2020

Révisions associées

Révision 219344e0 - 07 mai 2020 16:22 - Nicolas Roche

templatetags: manage decimals on mathematics filters (#41868)

Historique

#1 - 20 avril 2020 10:00 - Nicolas Roche

- Lié à Development #27709: Filtres Django pour les opérations mathématiques ajouté

#2 - 20 avril 2020 10:11 - Nicolas Roche

- Fichier 0001-templatetags-manage-decimals-on-mathematics-filters-.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#3 - 20 avril 2020 10:15 - Frédéric Péters

Ça casse le |add pour les usages non-arithmétiques, qu'on a par exemple dans des intégrations graphiques,

```
url({% asset_url "wcs:category:logo-active:"|add:cell.category_reference %});
```

#4 - 20 avril 2020 12:37 - Nicolas Roche

J'ai essayé de mixer les deux approches (pour avoir le même comportement que sur wcs),

- pour éviter que le add de django accepte 0.3 et le convertisse en 0
- pour convertir 'not a number' en 0 si l'autre terme est convertible en décimal

mais je réalise que l'on pourrait aussi vouloir concaténer un nombre à une chaîne, ce qui ne fonctionnera plus (ex: 'version' + '0.3').

Peut-être donner un nouveau nom comme "addition" pour l'addition arithmétiques ?

J'ai l'impression que de renommer "add" en "concat" dans les intégrations graphiques, ne suffirait pas puisque d'autres occurrences peuvent exister en base : cellules prototype json...

#5 - 20 avril 2020 15:18 - Nicolas Roche

- Fichier 0001-templatetags-manage-decimals-on-mathematics-filters-.patch ajouté

Dans ce patch j'ai opté pour 'sum' (à la place de 'add').
Si c'est validé, je me dit qu'il faudrait l'ajouter à wcs dans l'optique qu'il remplace le 'add' arithmétique sur le long terme.

#6 - 20 avril 2020 15:22 - Frédéric Péters

(non ne mettons pas un templatetag différent)

#7 - 20 avril 2020 16:30 - Nicolas Roche

- Fichier 0001-templatetags-manage-decimals-on-mathematics-filters-.patch ajouté

A présent, je tente le filtre arithmétique puis je me rabat sur le filtre natif en cas de conversion manquée (plutôt que d'utiliser la valeur 0).

Ce patch ne permet pas de concaténer des nombres.

Il introduit les régressions suivantes par rapports aux tests originaux (tous évalués à ") :

```
assert tpl.render(Context({'term1': 'not a number', 'term2': 1.2})) == '1.2'  
assert tpl.render(Context({'term1': 1.4, 'term2': "not a number"})) == '1.4'  
assert tpl.render(Context({'term1': '', 'term2': 3.5})) == '3.5'  
assert tpl.render(Context({'term1': 3.6, 'term2': ''})) == '3.6'  
assert tpl.render(Context({'term1': '', 'term2': ''})) == '0'  
assert tpl.render(Context({'term1': 0, 'term2': ''})) == '0'  
assert tpl.render(Context({'term1': '', 'term2': 0})) == '0'
```

#8 - 20 avril 2020 16:32 - Nicolas Roche

- Lié à Development #41699: Ajouter le filtre add ajouté

#9 - 20 avril 2020 16:32 - Nicolas Roche

- Lié à Development #41698: Ajouter le filtre subtract ajouté

#10 - 20 avril 2020 16:33 - Nicolas Roche

- Lié à Development #41697: Ajouter le filtre multiply ajouté

#11 - 20 avril 2020 16:33 - Nicolas Roche

- Lié à Development #41700: Ajouter le filtre divide ajouté

#12 - 21 avril 2020 08:48 - Frédéric Péters

- Sujet changé de Ajouter la conversion décimal sur les fitres matématiques par défaut à Ajouter la conversion décimal sur les fitres mathématiques par défaut

#13 - 22 avril 2020 12:23 - Nicolas Roche

Pour revenir sur <https://dev.entrouvert.org/issues/41868#note-4>,

je me demande où je devrais chercher pour voir où d'autre le filtre add serait utilisé par les utilisateurs, et en fin de compte j'ai l'impression que les gabarits ne sont pas souvent utilisés :

- les cellules prototype JSON
- les jsondatastores
- les requêtes csv et depuis peu opengis, arcgis

Est-ce que j'ai fait le tour ?

Auquel cas, je pourrais les adapter, ainsi que les intégrations graphiques afin d'intégrer le même opérateur que dans wc.

#14 - 22 avril 2020 12:28 - Frédéric Péters

Ce n'est pas réaliste du tout d'imaginer pouvoir tout modifier.

#15 - 22 avril 2020 12:36 - Thomas Noël

Nicolas Roche a écrit :

Pour revenir sur <https://dev.entrouvert.org/issues/41868#note-4>,

je me demande où je devrais chercher pour voir où d'autre le filtre add serait utilisé par les utilisateurs, et en fin de compte j'ai l'impression que les gabarits ne sont pas souvent utilisés :

- les cellules prototype JSON
- les jsondatastores
- les requêtes csv et depuis peu opengis, arcgis

Est-ce que j'ai fait le tour ?

Absolument à tout bout de champ dans wcs :)

La règle est assez simple : il ne faut pas toucher aux comportements existants.

Notre erreur dans wcs est d'avoir créé un "add" qui écrase celui qui existe dans Django. On n'y touchera pas, c'est pas très grave. Mais le add de Django est déjà utilisé par des usages de Combo, et donc on ne peut plus le remplacer.

Je propose, pour s'autoriser un "add" dans Combo : ne faire l'addition que si le membre de gauche est déjà un Decimal ; sans tenter de conversion. Et sinon, fallback vers le "add" natif de Django.

Cela permettra de faire « foo|decimal|add:bar », ça reste compréhensible.

#16 - 22 avril 2020 12:38 - Thomas Noël

Thomas Noël a écrit :

Je propose, pour s'autoriser un "add" dans Combo : ne faire l'addition que si le membre de gauche est déjà un Decimal ; sans tenter de conversion. Et sinon, fallback vers le "add" natif de Django.

Et je relis la doc de Django et c'est déjà ce que ça fait, en fait... il ne faut rien faire pour add :)

#17 - 22 avril 2020 12:49 - Frédéric Péters

(le truc qui pousse Nicolas ici c'est qu'on a une conversion de la virgule comme séparateur décimal)

#18 - 22 avril 2020 13:11 - Thomas Noël

Frédéric Péters a écrit :

(le truc qui pousse Nicolas ici c'est qu'on a une conversion de la virgule comme séparateur décimal)

Ok alors je n'ai pas trop d'avis s'il faut ou non conserver la magie de conversion. Si on veut conserver la magie, je serais pour ne faire le do_raise que sur le membre de gauche.

Aussi, le « from django.template.defaultfilters import add as native_add » est inutile, utiliser directement defaultfilters.add déjà disponible.

#19 - 22 avril 2020 16:11 - Nicolas Roche

- Fichier 0001-templatetags-manage-decimals-on-mathematics-filters-.patch ajouté

do_raise que sur le membre de gauche

oui, il n'y a plus que 3 régressions dans les tests importé depuis wcs.

#20 - 22 avril 2020 16:36 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Nicolas Roche a écrit :

do_raise que sur le membre de gauche

oui, il n'y a plus que 3 régressions dans les tests importé depuis wcs.

Ca me va bien.

Dans les tests il faut déplacer tous les « assert tmpl.render(Context({'term1': pas un nombre, ... » vers la section "native add" (et l'appeler "fallback to Django native add filter")

Et ack avec ce dernier mini détail.

#21 - 24 avril 2020 08:29 - Frédéric Péters

- Statut changé de Solution validée à Solution proposée

Attendre un peu, [#42062](#).

#23 - 04 mai 2020 16:11 - Nicolas Roche

- Lié à [Development #42062](#): permettre les manipulation de chaine avec `|add` ajouté

#24 - 06 mai 2020 09:49 - Frédéric Péters

Règles de [#42062](#) à appliquer ici. (si ce n'est déjà le cas)

#25 - 06 mai 2020 10:53 - Nicolas Roche

- Fichier `0001-templatetags-manage-decimals-on-mathematics-filters-.patch` ajouté

(Thomas, pour simplifier j'ai également appliqué les mêmes tests)

#26 - 06 mai 2020 16:35 - Thomas Noël

Il n'y a pas de notion de "unlazy" dans Django, c'est un truc lié à w.c.s., tu peux les retirer ici.

Le `parse_decimal` pourrait retourner `None` en cas d'impossibilité de conversion. Alors dans `decimal`, tu ferais « `value = parse_decimal(value) or Decimal(0)` ». Et tu éviterais tout un tas de `try/except` dans le `add`:

```
...
term1_decimal = parse_decimal(term1)
term2_decimal = parse_decimal(term2)

if term1_decimal is not None and term2_decimal is not None:
    return term1_decimal + term2_decimal
if term1 == '' and term2_decimal is not None:
    return term2_decimal
if term2 == '' and term1_decimal is not None:
    return term1_decimal
return defaultfilters.add(term1, term2)
```

#27 - 06 mai 2020 17:06 - Nicolas Roche

Il n'y a pas de notion de "unlazy" dans Django

aïe, j'ai honte.

Le `parse_decimal` pourrait retourner `None` en cas d'impossibilité de conversion.

bof, ce même code est déjà poussé sur wcs et ça casse pas mal l'existant, ne serait-ce que parce que les autres opérateurs utilisent aussi `parse_decimal()`.

```
def test_decimal_templatetag():
    tpl = Template('{{ plop|decimal }}')
> assert tpl.render(Context({'plop': 'toto'})) == '0'
E     AssertionError: assert 'None' == '0'
E         - 0
E         + None
```

Donc dites-moi si je refais un passage sur [#42062](#).

#28 - 06 mai 2020 17:20 - Frédéric Péters

Le `parse_decimal` pourrait retourner `None` en cas d'impossibilité de conversion.

bof, ce même code est déjà poussé sur wcs et ça casse pas mal l'existant

Très compliqué pour moi de comprendre si "ça casse" parle du code qui a été poussé dans w.c.s., et si "l'existant" parle de code/conditions/gabarits qui y existent; ou si "ça casse" est plutôt une réponse "non" à Thomas, que sa proposition "casserait" "un existant" (même inconnue ici).

#29 - 06 mai 2020 17:43 - Thomas Noël

Le code dans wcs a sa vie et son existant, qui nous a obligé à jouer les affaires autrement, laissons-le tranquille, n'y touchons plus, il roule. Mais ici parse_decimal est tout neuf donc y'a aucun soucis, et je trouve que le code de "add" avec juste des "if" serait bien plus lisible.

#30 - 06 mai 2020 18:34 - Nicolas Roche

J'étais parti du principe que l'on faisait juste du copier/coller des filtres entre combo et wcs, pour faciliter leur maintenance (c'est du moins le discours que j'ai entendu à plusieurs reprises).

Partant de là, comme le code de parse_decimal est utilisé sur wcs par les opérateurs arithmétiques d'une part, mais aussi dans les gabarits django, ma réponse était "plutôt non", si l'on veut conserver la gestion transversale (ou bête) du code des filtres entre les deux briques.

Et en complément induit dans ma réponse, je pointais qu'il me faudra également retoucher :

- decimal()
- subtract()
- multiply()
- divide()
- ceil()
- floor()
- abs()

pour lesquelles parse_decimal() avait été optimisé, mais aussi peut-être un jour (si on veut avoir le même code dans les 2 briques) :

- add_days()
- add_hours()

J'avertis que l'on a (selon moi) changé de position et demande un avis tranché de votre part.

#31 - 06 mai 2020 18:44 - Frédéric Péters

J'étais parti du principe que l'on faisait juste du copier/coller des filtres entre combo et wcs, pour faciliter leur maintenance (c'est du moins le discours que j'ai entendu à plusieurs reprises).

On ne peut pas faire de copié/collé il y a tout le concept "lazy" qui existe dans wcs et pas dans combo. Si ça a été dit, faut l'oublier, ça n'est pas possible.

#32 - 06 mai 2020 19:10 - Nicolas Roche

- Statut changé de Solution proposée à En cours

#33 - 07 mai 2020 01:04 - Thomas Noël

Nicolas Roche a écrit :

J'étais parti du principe que l'on faisait juste du copier/coller des filtres entre combo et wcs, pour faciliter leur maintenance (c'est du moins le discours que j'ai entendu à plusieurs reprises).

Oui c'est vrai, on dit ça... mais la réalité nous rattrape ici. Dans w.c.s. on avait un existant sur parse_decimal qu'on a évité de bouleverser. Sur Combo on les coudées franches, et je préférerais alors un add plus "joli".

Mais déjà, notons qu'un truc chouette ça va être de copier/coller les tests. Vu qu'ils couvrent tout ce qui nous intéresse, c'est déjà super.

#34 - 07 mai 2020 09:35 - Nicolas Roche

- Fichier 0001-templatetags-manage-decimals-on-mathematics-filters-.patch ajouté

- Statut changé de En cours à Solution proposée

Oui, les tests sont identiques des deux côtés.

#35 - 07 mai 2020 12:01 - Thomas Noël

Nicolas Roche a écrit :

Oui, les tests sont identiques des deux côtés.

Cool.

Bon, je vais t'entendre soupirer "rahhh ces relecteurs...", mais si on fait :

```
def parse_decimal(value, default=Decimal(0)):
```

```
...
except (ArithmeticError, TypeError):
    return default
```

alors dans "add" on utilisera des « parse_decimal(..., default=None) », et pour les autres opérateurs le code sera, ô victoire, celui de w.c.s :)

(et ensuite vraiment je pense que ça va être un ack, oui oui)

#36 - 07 mai 2020 15:31 - Nicolas Roche

- Fichier 0001-templatetags-manage-decimals-on-mathematics-filters-.patch ajouté

(et ensuite vraiment je pense que ça va être un ack, oui oui)

chiche !

#37 - 07 mai 2020 16:15 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Na !:-)

#38 - 07 mai 2020 16:23 - Nicolas Roche

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

```
commit 219344e0dbaa8f97f8a350bee1487ba22ddf981a
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Thu May 7 15:10:10 2020 +0200
```

```
templatetags: manage decimals on mathematics filters (#41868)
```

#39 - 07 mai 2020 17:16 - Frédéric Péters

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

#40 - 08 mai 2020 09:14 - Nicolas Roche

- Lié à Bug #42650: Erreur sur le filtre add ajouté

#41 - 08 mai 2020 10:15 - Nicolas Roche

- Lié à Development #42653: Simplification du filtre add ajouté

Fichiers

0001-templatetags-manage-decimals-on-mathematics-filters-.patch	7,46 ko	20 avril 2020	Nicolas Roche
0001-templatetags-manage-decimals-on-mathematics-filters-.patch	7,41 ko	20 avril 2020	Nicolas Roche
0001-templatetags-manage-decimals-on-mathematics-filters-.patch	7,79 ko	20 avril 2020	Nicolas Roche
0001-templatetags-manage-decimals-on-mathematics-filters-.patch	7,72 ko	22 avril 2020	Nicolas Roche
0001-templatetags-manage-decimals-on-mathematics-filters-.patch	9,38 ko	06 mai 2020	Nicolas Roche
0001-templatetags-manage-decimals-on-mathematics-filters-.patch	9,06 ko	07 mai 2020	Nicolas Roche
0001-templatetags-manage-decimals-on-mathematics-filters-.patch	8,8 ko	07 mai 2020	Nicolas Roche