

w.c.s. - Development #42062

permettre les manipulation de chaine avec |add

23 avril 2020 20:13 - Frédéric Péters

Statut:	Fermé	Début:	23 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			
On en parlait côté Combo mais voilà via #42029 que le besoin est également exprimé pour w.c.s.			
Demandes liées:			
Lié à Combo - Development #41868: Ajouter la conversion décimal sur les filtre...		Fermé	20 avril 2020
Lié à w.c.s. - Development #42653: Simplification du filtre add		Rejeté	08 mai 2020

Révisions associées

Révision 942afd8e - 06 mai 2020 10:05 - Nicolas Roche

templatetags: manage string concatenation on add filter (#42062)

Historique

#1 - 23 avril 2020 23:57 - Thomas Noël

Ah zut.

La logique actuelle dans Combo est : « on tente de faire une addition si le membre de gauche est un décimal ou est convertible en décimal, sinon on fait le add de Django »

Le risque est de faire planter les usages actuels de add sur une variable non existante ou égale à None (des choses comme {{ form_var_nb_mails_envoyes_Stat_raw|add:"1" }} sur Chambéry)

J'ai envie de conserver la magie et de ne pas imposer le "|decimal|add".

Du coup, je propose cette logique pour gauche|add:droite :

- si gauche est convertible en décimal, alors on converti "de force" droite en décimal (avec échec = 0) et on additionne
- si droite est convertible en décimal, alors on tente de convertir le gauche en décimal : si ça marche, on additionne. Si le gauche est vide ou inexistant, il vaut 0 et on additionne.
- dans les autres cas : on bascule vers le add de Django

Il restera le cas de deux variables inexistantes ou vides dont l'addition valait 0... ça ne sera plus vrai, mais ça reverra le vide, alors je considère que "ça va".

(et si c'est cette logique qu'on adopte, il faudra aussi la renvoyer vers Combo).

#2 - 24 avril 2020 07:59 - Frédéric Péters

Je serais, sans me soucier de la position des termes,

- nombre et nombre → addition
- nombre et None → nombre (+ 0)
- nombre et chaîne vide → nombre (+ 0)
- nombre et autre chose → django add
- None et None → None
- chaîne et None → chaîne (+ "")
- autre chose et autre chose → django add

en définissant comme nombre ce qui sort sans erreur de `Decimal(value).quantize(Decimal("1.0000")).normalize()` (ex: "12,3").

Et j'oublie peut-être des situations.

#3 - 24 avril 2020 10:54 - Nicolas Roche

- Statut changé de Nouveau à En cours

- Assigné à mis à Nicolas Roche

```
          number2  None  ''  other2
number1  n1+n2     n1   n1   django_add
None          None  ''  other2
''           ''    ''  other2
autre                               django_add
```

Il manque seulement : "" et None → ""

#4 - 24 avril 2020 11:30 - Nicolas Roche

- Fichier 0001-templatetags-manage-string-concatenation-on-add-filt.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

(Le filtre add Django retourne "" si on l'utilise avec None.)

#5 - 24 avril 2020 14:35 - Thomas Noël

Le filtre add Django retourne "" si on l'utilise avec None.

Mais tu n'as pas mis "native" dans la matrice pour ce cas, semble-t-il.

Aussi, quand un terme est "?" et que l'autre est None, tu renvoies vers onNone. Il faut plutôt faire du natif dans ce cas, je crois.

Et je ne suis pas très à l'aise avec l'idée de répondre "None" dans None+None, je préférerais qu'on réponde "". Pourquoi pensais-tu mieux de répondre None, Frédéric ?

#6 - 24 avril 2020 15:00 - Frédéric Péters

Je n'ai pas de souvenir particulier d'exemple me menant à ça, c'était il y a plusieurs heures déjà...

#7 - 24 avril 2020 15:49 - Thomas Noël

Je préfère que None+None renvoie "" comme le fait Django par défaut.

En reprenant les règles de Frédéric qui me vont bien :

- nombre et nombre → addition
- nombre et None → nombre (+ 0)
- nombre et chaîne vide → nombre (+ 0)
- chaîne et None → chaîne (+ "")

et "tout le reste" via django add, dans ce que listait Frédéric :

- nombre et autre chose → django add
- None et None → django add
- autre chose et autre chose → django add

On voit qu'avec cette logique, les None peuvent être remplacés par "" avant les tests, et il ne reste que :

- nombre et nombre → addition
- nombre et chaîne vide → nombre (+ 0)

Dans les tests, il faudrait aussi vérifier ce que tout ça va donner avec des variables non existantes à droite comme à gauche (il faudrait qu'elles soient considérées comme vides).

Nicolas, la notion de matrice est jolie, mais si essaye de voir quelques instants si on ne peut pas écrire ça sous une suite de "if" plus parlante, sait-on jamais.

Pour info le code de Django:

```
#####
# INTEGERS      #
#####

@register.filter(is_safe=False)
def add(value, arg):
```

```
"""Add the arg to the value."""
try:
    return int(value) + int(arg)
except (ValueError, TypeError):
    try:
        return value + arg
    except Exception:
        return ''
```

(ouaip, c'est rangé dans le code comme faisant partie des additions sur des entiers, amusant)

#8 - 24 avril 2020 20:14 - Nicolas Roche

- Fichier 0003-sms-add-endpoint-to-send-sms-asynchronously-21465.patch ajouté
- Fichier 0002-sms-factorise-sms-code-on-parameters-21465.patch ajouté
- Fichier 0001-sms-add-SMS-class-to-store-sms-parameters-21465.patch ajouté

#9 - 24 avril 2020 20:15 - Nicolas Roche

- Fichier 0001-sms-add-SMS-class-to-store-sms-parameters-21465.patch supprimé

#10 - 24 avril 2020 20:15 - Nicolas Roche

- Fichier 0002-sms-factorise-sms-code-on-parameters-21465.patch supprimé

#11 - 24 avril 2020 20:15 - Nicolas Roche

- Fichier 0003-sms-add-endpoint-to-send-sms-asynchronously-21465.patch supprimé

#12 - 24 avril 2020 20:16 - Nicolas Roche

- Fichier 0001-templatetags-manage-string-concatenation-on-add-filt.patch ajouté

(désolé pour la fausse manip)

#13 - 04 mai 2020 12:33 - Frédéric Péters

fail-back → fallback.

both termes → both terms.

Dans les tests, c'est bizarre d'avoir des lignes qui disparaissent et ne pas savoir ce qu'elles deviennent, genre :

```
- assert tmpl.render({'term1': 'not a number', 'term2': 1.2}) == '1.2'
```

je préférerais que la ligne reste, avec son résultat différent. (pareil pour les autres).

Manquent des tests avec la virgule comme séparateur décimal.

#14 - 04 mai 2020 16:11 - Nicolas Roche

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

#15 - 04 mai 2020 17:16 - Nicolas Roche

- Fichier 0001-templatetags-manage-string-concatenation-on-add-filt.patch ajouté

#16 - 06 mai 2020 09:48 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

#17 - 06 mai 2020 10:06 - Nicolas Roche

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

```
commit 942afd8e9f016dc02b4c4cd8fa4c6922517227ec
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Fri Apr 24 10:50:17 2020 +0200
```

```
templatetags: manage string concatenation on add filter (#42062)
```

#18 - 06 mai 2020 22:16 - Frédéric Péters

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

#19 - 08 mai 2020 10:15 - Nicolas Roche

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

Fichiers

0001-templatetags-manage-string-concatenation-on-add-filt.patch	5,46 ko	24 avril 2020	Nicolas Roche
0001-templatetags-manage-string-concatenation-on-add-filt.patch	9,97 ko	24 avril 2020	Nicolas Roche
0001-templatetags-manage-string-concatenation-on-add-filt.patch	10,9 ko	04 mai 2020	Nicolas Roche