

w.c.s. - Development #30852

dé-lazyfication des variables du formulaire dans le filtre django 'decimal'

22 février 2019 17:53 - Nicolas Roche

Statut:	Fermé	Début:	22 février 2019
Priorité:	Normal	Echéance:	
Assigné à:	Nicolas Roche	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		

Description

recommandations de Thomas liées à [#27709](#) pour avoir un code plus léger :

- 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.

Révisions associées

Révision e4ee1a90 - 25 février 2019 10:16 - Nicolas Roche

templatetags: beautify decimal filter's code (#30852)

Historique

#1 - 22 février 2019 18:03 - Thomas Noël

J'en suis même à me dire que `parse_decimal` pourrait toujours renvoyer `Decimal(0)` quand il n'arrive à rien lire... Ca pourrait être encore plus simple en code et en compréhension (mon idée étant que quelqu'un qui utilise le filtre `form_var_foo|decimal` pourra mieux comprendre que ça renvoie 0 plutôt que le vide : «au moins le filtre a marché, c'est juste le `form_var_foo` qui est pas bon»)

#2 - 22 février 2019 18:28 - Nicolas Roche

- Fichier `0001-templatetags-beautify-decimal-filter-s-code-30852.patch` ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Oui, c'est plus cohérent.

Cela tranche pour les résultat "magiques" de la multiplication et de la division en cas de valeurs vides ou bidons ([#27709](#)) :

- la multiplication renverra 0
- la division avec un mauvais dénominateur renverrait la chaîne vide
- sinon une division avec un mauvais numérateur renverrait 0

#3 - 23 février 2019 00:06 - Thomas Noël

Détail sur le début du patch :

```
assert tpl.render({'plop': 12345.678}) == '12345.678'
- assert tpl.render({'plop': None}) == ''
  assert tpl.render({'plop': 0}) == '0'
+ assert tpl.render({'plop': 'x'}) == '0'
+ assert tpl.render({'plop': None}) == '0'
```

- le test avec `'plop': 'x'` est inutile, c'est le même que `'plop': 'toto'`
- essaye de ne pas changer l'ordre des lignes pour simplifier la vie du relecteur, le `'plop': None` se retrouve déplacé ici sans raison.

Bref, on devrait plutôt avoir :

```
assert tpl.render({'plop': 12345.678}) == '12345.678'
- assert tpl.render({'plop': None}) == ''
+ assert tpl.render({'plop': None}) == '0'
  assert tpl.render({'plop': 0}) == '0'
```

ce qui fait gagner 10 secondes de vie au relecteur :)

Pour le reste ça me va bien, tu pourras pousser avec cette mini correction du test.

Ensuite il va falloir retourner sur le patch des opérateurs mathématiques de [#27709](#). Mais on va pouvoir y faire du des trucs comme `return parse_decimal(term1) + parse_decimal(term2)`, chic alors !

#4 - 25 février 2019 10:22 - Nicolas Roche

- *Statut changé de Solution proposée à Résolu (à déployer)*

```
commit e4ee1a902b03d768a2e7c4831fc3ab3df293fb0f
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date:   Fri Feb 22 17:58:27 2019 +0100
```

```
templatetags: beautify decimal filter's code (#30852)
```

#5 - 25 février 2019 13:52 - Thomas Noël

Je le dis ici, je n'avais pas "validé" la solution plus haut donc tu aurais éventuellement pu passer par une nouvelle proposition, mais finalement ça va, ce que je demandais de modifier l'a bien été, et tu as poussé ce qu'il faut, ça passe pour cette fois ;)

#6 - 25 février 2019 13:53 - Thomas Noël

ah ben non, j'avais explicitement écrit "tu pourras pousser avec cette mini correction du test." donc j'ai tort, c'est toi qui a bien fait ! :)

#7 - 25 février 2019 17:15 - Frédéric Péters

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

Fichiers

0001-templatetags-beautify-decimal-filter-s-code-30852.patch	3,47 ko	22 février 2019	Nicolas Roche
--	---------	-----------------	---------------