

w.c.s. - Bug #49948

Les liste à choix multiple sont interprétés comme des chaînes dans les gabarits utilisé dans les paramètres des appels webservices

07 janvier 2021 14:21 - Nicolas Roche

Statut:	Fermé	Début:	07 janvier 2021
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Les {{ xxx_raw }} d'une liste à choix multiple semblent retourner une chaîne '[xxx]' plutôt qu'une liste ['xxx'] (contrairement à ce que remonte l'inspecteur)

Sur un appel [webservice](#), je décompose les élément d'une liste à choix multiple en une chaîne séparée par des caractères ':', avec par exemple ce gabarit : {{ form_var_adultes_raw | join:":" }}

En utilisant ce gabarit dans l'[inspecteur](#) j'ai bien la chaîne attendue :

```
435929:439240
```

Mais si je vais voir les [logs dans passerelle](#) :

```
exception individusConcernes: "[:':4:3:5:9:2:9:':,,: ':4:3:9:2:4:0:':]:\ue001..." does not match '^[0-9 :]+$',
```

Localement, j'ai rejoué une demande qui était passée auparavant. J'en déduis qu'il s'agit d'une régression. (mais sinon, peut-être que je devrais simplement utiliser la virgule comme séparateur dans le connecteur ?)

Demandes liées:

Lié à w.c.s. - Development #49970: {% with versus données complexes	Fermé	08 janvier 2021
Lié à w.c.s. - Development #49971: {% for versus données complexes	Fermé	08 janvier 2021

Révisions associées

Révision f356a182 - 11 janvier 2021 16:19 - Frédéric Péters

misc: revamp template complex data handling to monkeypatch last step (#49948)

Historique

#1 - 07 janvier 2021 14:28 - Frédéric Péters

\ue001 montre qu'il y a une espèce de n'importe quoi autour des données complexes.

Mais tous les liens mis à part, ton ticket, c'est un appel webservice, avec {{ form_var_adultes_raw | join:":" }} comme paramètre, et un résultat où tu voudrais "A:B:C" mais où tu reçois autre chose (un peu confus dans ma lecture de ton deuxième bloc de code.

#2 - 07 janvier 2021 14:43 - Thomas Noël

Oui, on a un form_var_truc_raw qui est une liste [1,2].

Mais quand on fait un {{ form_var_truc_raw|join:":" }} c'est comme si c'était liste était déjà transformée en string car on obtient [:1,;2]:

#3 - 07 janvier 2021 14:51 - Frédéric Péters

Si vous voulez,

```
--- a/tests/test_workflows.py
+++ b/tests/test_workflows.py
@@ -2244,6 +2244,7 @@ def test_webservice_with_complex_data(http_requests, pub):
     'ezt_item_raw': '[form_var_item_raw]',
     'items_raw': '{{ form_var_items_raw }}',
```

```

    'ezt_items_raw': '[form_var_items_raw]',
+   'joined_item_raw': '{{ form_var_item_raw|join:"|" }}',
  }
  pub.substitutions.feed(formdata)
  with get_publisher().complex_data():
@@ -2260,6 +2261,7 @@ def test_webservice_with_complex_data(http_requests, pub):
    'ezt_item_raw': 'a',
    'items_raw': ['a', 'b'],
    'ezt_items_raw': ['a', 'b'],
+   'joined_item_raw': 'a|b',
  }

```

sinon je regarde demain.

#4 - 07 janvier 2021 21:57 - Frédéric Péters

- Assigné à mis à Frédéric Péters

#5 - 08 janvier 2021 07:28 - Frédéric Péters

- Fichier 0001-misc-monkeypatch-FilterExpression-to-handle-complex-.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Obligé de passer par monkeypatcher le traitement des filtres et ça me révèle qu'il pourrait y avoir d'autres affaires du même genre (pas testé mais je pense à des trucs autour de {% with }, ou { for %}, même); là où on active les données complexes on doit rester sur des gabarits simples, donc ça doit en pratique aller.

#6 - 08 janvier 2021 10:07 - Nicolas Roche

Je valide d'un point de vue fonctionnel : le patch fonctionne bien dans mon cas d'usage (par contre je ne saurais pas donner mon avis sur le patch).

#7 - 08 janvier 2021 10:54 - Thomas Noël

Aïe.

Frédéric Péters a écrit :

là où on active les données complexes on doit rester sur des gabarits simples, donc ça doit en pratique aller.

Si « for » ne marche pas, ça me paraît très délicat. Je ne vois pas trop comment expliquer aux gens de ne pas l'utiliser, sachant que ceux qui configurent les appels webservices ou les attributions de données de traitement connaissent for voire with.

Et c'est confirmé que ça ne passe pas :

```

--- a/tests/test_workflows.py
+++ b/tests/test_workflows.py
@@ -2214,6 +2214,7 @@ def test_webservice_with_complex_data(http_requests, pub):
    'items_raw': '{{ form_var_items_raw }}',
    'ezt_items_raw': '[form_var_items_raw]',
    'joined_items_raw': '{{ form_var_items_raw|join:"|" }}',
+   'for_items_raw': '{% for item in form_var_items_raw %}{{item}}-{% endfor %}',
  }
  pub.substitutions.feed(formdata)
  with get_publisher().complex_data():
@@ -2231,6 +2232,7 @@ def test_webservice_with_complex_data(http_requests, pub):
    'items_raw': ['a', 'b'],
    'ezt_items_raw': ['a', 'b'],
    'joined_items_raw': 'a|b',
+   'for_items_raw': 'a-b-',
  }

```

se termine en :

```

E Differing items:
E {'for_items_raw': "[-'a'-,- '-b'-]-\ue00b-"} != {'for_items_raw': 'a-b-'}

```

#8 - 08 janvier 2021 10:56 - Frédéric Péters

Ça peut être un autre ticket ? Pendant ce temps je fais l'inventaire de la prod pour voir.

#9 - 08 janvier 2021 11:12 - Frédéric Péters

- Lié à [Development #49970](#): {% with versus données complexes ajouté

#10 - 08 janvier 2021 11:14 - Frédéric Péters

- Lié à [Development #49971](#): {% for versus données complexes ajouté

#11 - 08 janvier 2021 11:15 - Frédéric Péters

Voilà deux autres tickets puisque l'un se trouve présent en prod et l'autre pas.

#12 - 09 janvier 2021 13:49 - Frédéric Péters

- Fichier `0001-misc-revamp-template-complex-data-handling-to-monkey.patch` ajouté

J'étais content à ne pas avoir du monkey-patcher le système de templates de Django pour faire la gestion des données complexes mais se révèle donc que ça ne va pas tenir; plutôt que raccrocher les bouts, le patch attaché reprend du début pour changer l'approche : garder les données structurées tout du long et laisser la conversion avec ajout marqueur à la toute fin, ce qui signifie patcher uniquement le `render_value_in_context` de django.

Ça marche ainsi dans le patch attaché, où on peut donc voir :

- le monkey-patch sur `django.template.base.render_value_in_context`
- la suppression de la partie `str_value` de `cache_complex_data`, qui servait pour le cas particulier des blocs de champs (où `{{ form_var_le_bloc }}` pouvait à la fois être la donnée complexe, ou la représentation "digest_template" du bloc).
- la modification à `get_cached_complex_data` pour gérer les variables lazy
- la suppression dans `wcs/variables.py` de tout ce qui avait été ajouté pour la gestion des données complexes
- des tests avec des `{% for` et des `{% with`
- la modification du résultat du test `ezt`, qui ne gère plus les données complexes (`'ezt_items_raw': repr(['a', 'b'])`) (vu que ça ne se passe plus au niveau des variables mais au niveau du rendu final django, ça ne s'applique plus automatiquement à l'`ezt`).
- comme donc l'`ezt` est ignoré, la partie nettoyage qui avait été ajoutée dans [#49679](#) peut également être retirée.

À part le test `ezt` mentionné ci-dessus, tous les autres tests fonctionnent, donc pour moi c'est assez sûr comme changement.

#13 - 11 janvier 2021 15:13 - Thomas Noël

Quelque part, au final, c'est plus clair ainsi.

#14 - 11 janvier 2021 15:13 - Thomas Noël

- Statut changé de *Solution proposée* à *Solution validée*

#15 - 11 janvier 2021 16:20 - Frédéric Péters

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

Oui, au final c'est mieux ainsi,

```
commit f356a18261895597a438be06dcc725a5267a2f83
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Thu Jan 7 23:18:47 2021 +0100
```

```
misc: revamp template complex data handling to monkeypatch last step (#49948)
```

#16 - 11 janvier 2021 17:16 - Frédéric Péters

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

Fichiers

<code>0001-misc-monkeypatch-FilterExpression-to-handle-complex-.patch</code>	2,54 ko	08 janvier 2021	Frédéric Péters
<code>0001-misc-revamp-template-complex-data-handling-to-monkey.patch</code>	6,48 ko	09 janvier 2021	Frédéric Péters