

## w.c.s. - Bug #10447

### chaines unicode dans la vue /inspect

25 mars 2016 18:49 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	25 mars 2016
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Benjamin Dauvergne	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>	v1.40	<b>Planning:</b>	
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
Exception: <pre>type = '&lt;type 'exceptions.UnicodeDecodeError'&gt;', value = ''ascii' codec can't decode byte 0xc2 i n position 42: ordinal not in range(128)'</pre>			
Stack trace (most recent call first): File <code>"/usr/lib/python2.7/dist-packages/wcs/backoffice/management.py"</code> , line 1824, in inspect 1822         r += htmltext('</div>') 1823         r += htmltext('</ul>') > 1824         return r.getvalue() 1825 1826 ... 'sas_response_data_adult1_address_street_text': 'MODIGLIANI', u'form_var_address_city_post_code': u'94140', 'sas_response_data_family_address_street_text': 'MODIGLIANI', 'form_user_var_address'			
<b>Demandes liées:</b>			
Dupliqué par w.c.s. - Bug #10539: crash encodage sur /inspect		<b>Rejeté</b>	<b>04 avril 2016</b>

#### Révisions associées

##### Révision b5a493b9 - 05 avril 2016 16:06 - Benjamin Dauvergne

misc: encode data source value using qommon.misc.json\_encode\_helper (#10447)

It prevents data sources from outputting unicode strings.

##### Révision efbf9150 - 05 avril 2016 16:06 - Benjamin Dauvergne

backoffice: prevent non utf-8 strings from breaking /inspect (#10447)

##### Révision d88d726e - 06 avril 2016 17:23 - Frédéric Péters

trivial: use %r in inspect string (#10447)

#### Historique

##### #1 - 25 mars 2016 21:00 - Benjamin Dauvergne

- Fichier `0001-decode-remote-data-source-using-qommon.misc.json_loa.patch` ajouté

- Assigné à mis à Benjamin Dauvergne

- Patch proposed changé de Non à Oui

##### #2 - 25 mars 2016 21:01 - Benjamin Dauvergne

C'est bizarre qu'on ait pas eu de souci avant quand même.

##### #3 - 25 mars 2016 21:42 - Frédéric Péters

```
print 'structured_items', structured_items
```

Je ne suis pas sûr du `str()` forcé, le `json_loads`, il assure déjà que les chaines unicode se trouvent encodées en `str`. (il me semble) Et si c'est une `data_source` qui ne passe pas par le `json_loads`, le `site_encode()` est plus adapté que le `str()`.

À côté, il y a aussi besoin d'un patch, différent, pour que la page `/inspect` ne plante pas.

Et ça serait sans doute bien d'avoir un test ou l'autre.

#### #4 - 25 mars 2016 23:19 - Benjamin Dauvergne

Frédéric Péters a écrit :

[...]

Je ne suis pas sûr du str() forcé, le json\_loads, il assure déjà que les chaînes unicode se trouvent encodées en str. (il me semble) Et si c'est une data\_source qui ne passe pas par le json\_loads, le site\_encode() est plus adapté que le str().

Non il ne convertit que les chaînes en chaînes, dans les tests il y a des sources JSON avec des entiers qui plantent sans le str() forcé (et ne plantez pas avec le site\_encode). get\_items() est un cas particulier où on ne veut que des chaînes, le str() forcé me semble donc pertinent.

À côté, il y a aussi besoin d'un patch, différent, pour que la page /inspect ne plante pas.

Yep.

Et ça serait sans doute bien d'avoir un test ou l'autre.

Un test complet avec une datasource, un formulaire avec un champ item issu de la datasource et un accès sur /inspect ?

#### #5 - 25 mars 2016 23:27 - Benjamin Dauvergne

- Fichier 0001-decode-remote-data-source-using-qommon.misc.json\_loa.patch ajouté

- Fichier 0002-prevent-non-utf-8-strings-from-breaking-inspect-1044.patch ajouté

#### #6 - 28 mars 2016 03:31 - Benjamin Dauvergne

- Fichier 0002-prevent-non-utf-8-strings-from-breaking-inspect-1044.patch ajouté

Patch 2 amélioré.

#### #7 - 29 mars 2016 09:36 - Frédéric Péters

Un test complet avec une datasource, un formulaire avec un champ item issu de la datasource et un accès sur /inspect ?

Concernant le patch sur le /inspect, des tests passant sur les différentes situations traitées. (les branches du if et 2× try/except)

Pour le patch sur le data\_source, je dirais qu'un champ item alimenté par une data source produisant des chaînes unicode (et en écrivant ça je me dis qu'il faudrait plutôt une fonction générale qui serait tout aussi bien appelée pour la situation "formule python", plutôt que traiter uniquement le cas du json.

Patch 2 amélioré.

Il s'y est ajouté un "validate" au \_q\_export, sans objet.

#### #8 - 29 mars 2016 19:03 - Benjamin Dauvergne

- Fichier 0001-encode-data-source-value-using-qommon.misc.json\_enco.patch ajouté

- Fichier 0002-prevent-non-utf-8-strings-from-breaking-inspect-1044.patch ajouté

Frédéric Péters a écrit :

Un test complet avec une datasource, un formulaire avec un champ item issu de la datasource et un accès sur /inspect ?

Concernant le patch sur le /inspect, des tests passant sur les différentes situations traitées. (les branches du if et 2× try/except)

Pour le patch sur le data\_source, je dirais qu'un champ item alimenté par une data source produisant des chaînes unicode (et en écrivant ça je me dis qu'il faudrait plutôt une fonction générale qui serait tout aussi bien appelée pour la situation "formule python", plutôt que traiter uniquement le cas du json.

Tests ajoutés avec passage dans toutes les branches, à noter que ça ajoute une dépendance sur pyquery (accroché à webtest) pour les tests; dépendance que j'ai mise dans le tox.ini du ticket [#10473](#). Ça dépend aussi du ticket UTF-8 (quand on commence à faire des recherches de chaînes

accentués das le contenu des réponses, les problèmes se voient tout de suite), [#10472](#).

Patch 2 amélioré.

Il s'y est ajouté un "validate" au `_q_export`, sans objet.

Yep c'était un bout du [#10444](#) que je développe dans la même branche.

#### #9 - 30 mars 2016 10:22 - Benjamin Dauvergne

- Fichier `0002-prevent-non-utf-8-strings-from-breaking-inspect-1044.patch` ajouté

Le même avec un contournement du bug dans `htmltext()` découvert dans [#10473](#).

```
-         r += htmltext(' <span class="type">(%s)</span>') % type(v)
+         r += htmltext(' <span class="type">(%s)</span>') % repr(type(v))
```

#### #10 - 30 mars 2016 10:24 - Frédéric Péters

Et mettre `%r` dans la chaîne, plutôt ?

#### #11 - 30 mars 2016 10:36 - Benjamin Dauvergne

- Fichier `0002-prevent-non-utf-8-strings-from-breaking-inspect-1044.patch` ajouté

Jawort !

#### #12 - 04 avril 2016 11:36 - Frédéric Péters

- Dupliqué par Bug [#10539](#): crash encodage sur `/inspect` ajouté

#### #13 - 05 avril 2016 16:13 - Frédéric Péters

- Statut changé de Nouveau à Résolu (à déployer)

Merci, j'ai poussé.

```
commit ebbf9150689b42537b308893d8f9c5251216a3b7
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Fri Mar 25 23:25:39 2016 +0100
```

```
backoffice: prevent non utf-8 strings from breaking /inspect (#10447)
```

```
commit b5a493b941b161d19ce7c1b1cf2dae0bba20f2c6
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Fri Mar 25 19:46:49 2016 +0100
```

```
misc: encode data source value using qommon.misc.json_encode_helper (#10447)
```

```
It prevents data sources from outputting unicode strings.
```

#### #14 - 06 avril 2016 17:09 - Benjamin Dauvergne

T'as du pousser la mauvaise version ce bout s'est perdu:

```
diff --git a/wcs/backoffice/management.py b/wcs/backoffice/management.py
index cce82c9..09ed50a 100644
--- a/wcs/backoffice/management.py
+++ b/wcs/backoffice/management.py
@@ -1852,7 +1852,7 @@ class FormBackOfficeStatusPage(FormStatusPage):
     r += htmltext('<li>%s') % (k, k)
     r = htmltext(' <div class="value"><span>%s</span>') % safe(v)
     if not isinstance(v, basestring):
-        r += htmltext(' <span class="type">(%s)</span>') % repr(type(v))
+        r += htmltext(' <span class="type">(%r)</span>') % type(v)
     r += htmltext('</div>')
 r += htmltext('</ul>')
 return r.getvalue()
```

Mais je pense que ça ne nécessite pas un nouveau commit.

**#15 - 06 avril 2016 17:24 - Frédéric Péters**

J'ai quand même poussé la modif.

**#16 - 15 avril 2016 15:45 - Thomas Noël**

- *Version cible mis à v1.40*

**#17 - 01 mai 2016 20:53 - Frédéric Péters**

- *Statut changé de Résolu (à déployer) à Fermé*

**Fichiers**

---

0001-decode-remote-data-source-using-qommon.misc.json_loa.patch	2,18 ko	25 mars 2016	Benjamin Dauvergne
0001-decode-remote-data-source-using-qommon.misc.json_loa.patch	2,14 ko	25 mars 2016	Benjamin Dauvergne
0002-prevent-non-utf-8-strings-from-breaking-inspect-1044.patch	1,9 ko	25 mars 2016	Benjamin Dauvergne
0002-prevent-non-utf-8-strings-from-breaking-inspect-1044.patch	2,41 ko	28 mars 2016	Benjamin Dauvergne
0001-encode-data-source-value-using-qommon.misc.json_enco.patch	4,3 ko	29 mars 2016	Benjamin Dauvergne
0002-prevent-non-utf-8-strings-from-breaking-inspect-1044.patch	4,39 ko	29 mars 2016	Benjamin Dauvergne
0002-prevent-non-utf-8-strings-from-breaking-inspect-1044.patch	4,53 ko	30 mars 2016	Benjamin Dauvergne
0002-prevent-non-utf-8-strings-from-breaking-inspect-1044.patch	4,52 ko	30 mars 2016	Benjamin Dauvergne