w.c.s. - Bug #9786

erreur de selecteur jsonp avec des variables qui viennent d'une autre liste

27 janvier 2016 12:21 - Thomas Noël

Statut:	Fermé	**Début:**	27 janvier 2016
Priorité:	Haut	**Echéance:**	
Assigné à:	Thomas Noël	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:	v1.32		
Patch proposed:	Oui	**Planning:**	

Description

Depuis ce matin, signalé par Alfortville ~~je n'arrive pas à reproduire sur mes navigos locaux~~

Bogue sur la page "adresse de mon foyer" de
https://demarches2016.alfortville.fr/mon-dossier-famille/creation-de-mon-dossier-famille/

Le nom de la voie est une source JSONP avec [var_xxx] où xxx est une liste dans la même page, cette liste étant une source ID/value

La requête JSONP arrivant sur passerelle est :
/agoraplus/integration/address/communes/Alfortville/types-of-streets/Rue/?format=jsonp&....

Au lieu de :
/agoraplus/integration/address/communes/1/types-of-streets/11/?format=jsonp&....

```
<div class="SingleSelectHintWidget widget widget-required" id="var_address_type_of_street" data-va
luecontainerid="form_f44"><div class="title"><label for="form_f44">Type de voie<span class="requir
ed">*</span></label></div><div class="content"><select id="form_f44" name="f44">
<option value="0"></option>
<option value="1">Allée</option>
<option value="10">Résidence</option>
<option value="11">Rue</option>
<option value="12">Sente</option>

(...)

// construction du JSONP:

function url_replace_f45() {
    var url = $("#form_f45").data('select2').opts.wcs_base_url;
    selector = '#' + $('#var_address_city').data('valuecontainerid');
    url = url.replace('[var_address_city]', $(selector).val());
    selector = '#' + $('#var_address_type_of_street').data('valuecontainerid');
    url = url.replace('[var_address_type_of_street]', $(selector).val());
    $("#form_f45").data('select2').opts.ajax.url = url;
    $("#form_f45").data('select2').clear();
}
```

Révisions associées

Révision 4b1a8293 - 04 février 2016 18:23 - Frédéric Péters

misc: don't feed var_-prefixed variables into the substitution system (#9786)

They are only created for backward-compatibility when evaluating conditions and
interfere with variadic URLs that may be used for dynamic jsonp-based fields.

Historique

#1 - 27 janvier 2016 12:29 - Thomas Noël

Les navigateurs qui posent cette requête erronée :

- Mozilla/5.0 (Windows NT 6.1; WOW64; rv:42.0) Gecko/20100101 Firefox/42.0 12:26:15

- Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36

#2 - 27 janvier 2016 12:37 - Thomas Noël

Bogue trouvé :

- aller sur la page
- oublier un champ
- valider
- l'URL devient alors et pour toujours /agoraplus/integration/address/communes/Alfortville/types-of-streets/Rue/ parce que les [var_....] sont remplacées par les variables du formulaires, désormais existantes

#3 - 27 janvier 2016 12:41 - Thomas Noël

Dans qommon/form.py

```
1704         if '[' in self.url:
1705             vars = get_publisher().substitutions.get_context_variables()
1706             # skip variables that were not set (None)
1707             vars = dict((x, y) for x, y in vars.items() if y is not None)
1708             url = misc.get_variadic_url(self.url, vars, encode_query=False)
```

il faut aussi ignorer (skip) toutes les variables qui sont sur la page actuelle

#4 - 27 janvier 2016 14:16 - Thomas Noël

- Assigné à mis à Thomas Noël

#5 - 27 janvier 2016 16:50 - Thomas Noël

- Fichier 0001-don-t-staticly-replace-var_xxx-in-jsonp-url.patch ajouté

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

Zut, impossible à reproduire en local, je n'ai pas de "var_" dans les variables de substitution, je ne comprends pas encore pourquoi :-/

Voici tout de même ce que je propose : ne jamais remplacer les [var_xxx] dans l'URL, toujours considérer que ces variables peuvent être locales à la page. Si on veut "forcer" des variables qui viennent de la session ou du formulaire, il faut utiliser les [form...] et autres [user...], [session...], etc.

C'est un patch "pour avis" (puisque je ne suis pas parvenu à reproduire le soucis sur ma machine).

#6 - 27 janvier 2016 19:54 - Thomas Noël

Conditions pour reproduire :

- un formulaire avec une page contenant une condition (c'est ça qui ajoute les var_ dans les variables du contexte)
- un premier champ liste avec source de données JSON, varname "x"
- un second champ liste avec une source de données JSONP dont l'URL contient [var_x]
  En cas de page non valide, var_x se retrouve valué dans les variables du contexte, et l'URL du JSONP n'est plus dynamique ([var_x] y est remplacé par la valeur une fois pour toute).

C'est ce que corrige le patch ci-dessus, en remplaçant toujours les var_ de façon dynamique.

#7 - 27 janvier 2016 19:56 - Thomas Noël

- Description mis à jour

#8 - 27 janvier 2016 20:20 - Frédéric Péters

La correction ne me va pas trop, on n'a pas envie de maintenir deux jeux de variables et d'avoir à expliquer que parfois c'est var_ et parfois c'est form_var et quand exactement on ne sait pas trop.

Selon moi ce qu'il faut c'est pouvoir, dans le cas précis du jsonp, dire qu'on ne veut pas les variables de la page courante pas encore validée. Ce que je verrais c'est donc un paramètre exclude supplémentaire à get_context_variables, par lequel on pourrait passer une liste des sources de données à exclure; d'ensuite marquer ConditionVars (du evaluate de la page de condition) avec un identifiant et de passer celui-ci dans le nouveau paramètre exclude.

Les conditions étant établies c'est maintenant normalement facile d'ajouter un test.

#9 - 27 janvier 2016 20:32 - Thomas Noël

Frédéric Péters a écrit :

> La correction ne me va pas trop, on n'a pas envie de maintenir deux jeux de variables et d'avoir à expliquer que parfois c'est var_ et parfois c'est

> form_var et quand exactement on ne sait pas trop.
>
> Selon moi ce qu'il faut c'est pouvoir, dans le cas précis du jsonp, dire qu'on ne veut pas les variables de la page courante pas encore validée. Ce que je verrais c'est donc un paramètre exclude supplémentaire à get_context_variables, par lequel on pourrait passer une liste des sources de données à exclure; d'ensuite marquer ConditionVars (du evaluate de la page de condition) avec un identifiant et de passer celui-ci dans le nouveau paramètre exclude.

Exclure ConditionVars ne va pas exclure les variables de la page courante non validé, mais toutes les var_xx. Ce qui est une bonne chose, puisqu'elles n'étaient pas toujours là (il fallait être dans une page avec une condition).

Donc oui, si on peut nettoyer au lieu de tenter comme moi de conserver un fonctionnement tordu, c'est mieux. Je regarde à faire ça.

#10 - 27 janvier 2016 20:33 - Thomas Noël

- Patch proposed changé de Oui à Non

#11 - 27 janvier 2016 20:57 - Frédéric Péters

> Exclure ConditionVars ne va pas exclure les variables de la page courante non validé, mais toutes les var_xx. Ce qui est une bonne chose, puisqu'elles n'étaient pas toujours là (il fallait être dans une page avec une condition).

À d'autres endroits ça ne devrait de toute façon pas être utilisé, c'était seulement explicitement mis dans l'évaluation des conditions pour compabilité avec des versions antérieures.

L'autre endroit où des [var_...] apparaissent, c'est bien le jsonp mais même là ça devrait pouvoir être remplacé par des [form_var_...], ou alors par un nom tout autre, qui n'apportera pas confusion. Tiens, ça peut d'ailleurs peut-être être une autre piste pour ce ticket, ajouter au jsonp paramétrique la prise en compte d'un autre préfixe (en supplément du var_ à garder pour compatibilité), mais c'est un peu foutraque javascript, plus difficile à tester.

#12 - 28 janvier 2016 08:21 - Frédéric Péters

Après avoir dormi dessus, je me dis que ton approche initiale peut aller, et même de manière plus franche, dans cet esprit :

```
--- a/wcs/qommon/form.py
+++ b/wcs/qommon/form.py
@@ -1716,8 +1716,11 @@ $("#form_%(id)s").select2({

         if '[' in self.url:
             vars = get_publisher().substitutions.get_context_variables()
-            # skip variables that were not set (None)
-            vars = dict((x, y) for x, y in vars.items() if y is not None)
+            # skip variables that were not set (None) and backward-compatiblity
+            # variables (starting with var_) that conflicts with the variables
+            # that can be dynamically set by javascript on field changes.
+            vars = dict((x, y) for x, y in vars.items() if
+                    y is not None and not x.startswith('var_'))
             url = misc.get_variadic_url(self.url, vars, encode_query=False)
         else:
             url = self.url
```

(et peu importe l'option retenue, elle doit être mise en place aussi dans AutocompleteStringWidget)

#13 - 28 janvier 2016 08:29 - Frédéric Péters

Autre approche qui me semble très bien, ce serait de reprendre le evaluate_condition() du champ Page et de ne pas insérer dans ConditionVars les variables de compatibilité (var_), quelque chose comme ça : (en actualisant les commentaires)

```
--- a/wcs/fields.py
+++ b/wcs/fields.py
@@ -1346,18 +1346,19 @@ class PageField(Field):
         # and add them as form_var_xxx.
         from formdata import get_dict_with_varnames
         live_data = get_dict_with_varnames(formdef.fields, dict)
-        for k, v in live_data.items():
-            live_data['form_' + k] = v
+        form_live_data = dict(('form_' + x, y) for x, y in live_data.items())

         # and feed those new variables in the global substitution system, they
         # will shadow formdata context variables with their new "live" value,
         # this may be useful when evaluating data sources.
         class ConditionVars(object):
             def get_substitution_variables(self):
```

```
-                 return live_data
+                 return form_live_data

          get_publisher().substitutions.feed(ConditionVars())
          data = get_publisher().substitutions.get_context_variables()
+         data.update(live_data)
+         data.update(form_live_data)

          try:
              if eval(condition, get_publisher().get_global_eval_dict(), data):
```

#14 - 28 janvier 2016 17:47 - Thomas Noël

Cette dernière solution ; très bien vue !

#15 - 29 janvier 2016 10:12 - Thomas Noël

Thomas Noël a écrit :

> Cette dernière solution ; très bien vue !

Et finalement à reprendre ce patch, je n'en suis plus aussi sûr : ça m'ennuie un peu de changer le comportement, même «bizarre», du calcul des conditions. En fait, je n'arrive pas à percevoir les effets exacts de ne plus mettre les var_ dans le get_substitution_variables().

Et donc je pencherai plutôt le patch du commentaire #12 qui nettoie les var_

#16 - 29 janvier 2016 10:32 - Frédéric Péters

L'apparition des var_... dans les variables de substitution c'est une conséquence de #8272 et personne n'en dépend (je laisse le soin au lecteur de vérifier ça); je préfère vraiment l'option de corriger le problème à la source, qui est l'ajout de ces var_, plutôt qu'au moment où attention elles sont là mais il ne faut pas les consommer.

#17 - 04 février 2016 15:22 - Thomas Noël

- Version cible mis à v1.32

#18 - 04 février 2016 18:18 - Frédéric Péters

- Fichier 0001-misc-don-t-feed-var_-prefixed-variables-into-the-sub.patch ajouté

- Patch proposed changé de Non à Oui

#19 - 04 février 2016 18:21 - Thomas Noël

Ack (quoique y'a un petit # qui traine dans un commentaire, c'est un scandale)

#20 - 04 février 2016 18:24 - Frédéric Péters

- Statut changé de En cours à Résolu (à déployer)

Corrigé.

```
commit 4b1a829340b995e904fc443ea74f59a9a7e222bd
Author: Frédéric Péters <fpeters@entrouvert.com>
Date:   Thu Feb 4 18:15:49 2016 +0100

    misc: don't feed var_-prefixed variables into the substitution system (#9786)

    They are only created for backward-compatibility when evaluating conditions and
    interfere with variadic URLs that may be used for dynamic jsonp-based fields.
```

#21 - 12 février 2016 15:17 - Thomas Noël

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-don-t-staticly-replace-var_xxx-in-jsonp-url.patch	2,05 ko	27 janvier 2016	Thomas Noël
0001-misc-don-t-feed-var_-prefixed-variables-into-the-sub.patch	4,18 ko	04 février 2016	Frédéric Péters