

w.c.s. - Development #44123

Vue de traitement : meilleure présentation d'un champ liste à choix multiple

16 juin 2020 11:02 - Emmanuel Cazenave

Statut:	Fermé	Début:	16 juin 2020
Priorité:	Normal	Echéance:	
Assigné à:	Emmanuel Cazenave	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Dans #44106, il y a un constat de mauvaise lisibilité d'un champ liste à choix multiple dans la vue de traitement : "Merci de nous indiquer les dates" dans https://demarches-rouen.test.entrouvert.org/backoffice/management/centre-loisirs/21/ .			
(et je me dis naïvement qu'une présentation en bullet ou autre pourrait faire l'affaire).			
Demandes liées:			
Lié à Intégrations graphiques Publik - Development #52097: Styler les listes ...		Fermé	16 mars 2021
Lié à w.c.s. - Development #50532: Le résumé du formulaire ne permet pas de r...		Fermé	26 janvier 2021

Révisions associées

Révision 149c73db - 16 mars 2021 14:11 - Emmanuel Cazenave

fields: store_structured on ItemsField for every data source (#44123)

Révision d44e6155 - 16 mars 2021 14:11 - Emmanuel Cazenave

fields: display ItemsField using on summary page if possible (#44123)

Historique

#2 - 16 juin 2020 11:25 - Frédéric Péters

Dans le travail en cours sur les blocs de champs il y a un **kwargs ajouté de manière systématique aux méthodes get_view_value(), ça permettra de distinguer les situations mieux qu'aujourd'hui, d'avoir un rendu là dans la vue de traitement d'une demande différent de ce qui est affiché dans une colonne du tableau de traitement.

(ça peut déjà être possible aujourd'hui via la distinction get_view_value et get_short_view_value mais je déconseille d'avancer là-dessus avant les blocs de champs).

#3 - 26 janvier 2021 15:23 - Emmanuel Cazenave

- Assigné à mis à Emmanuel Cazenave

#4 - 25 février 2021 15:04 - Emmanuel Cazenave

- Fichier Screenshot_2021-02-25 Backoffice de Démarches - choix multiple - 1.png ajouté

Pour initier le truc, un patch naïf :

```
iff --git a/wcs/fields.py b/wcs/fields.py
index 91a55bd3..c479864e 100644
--- a/wcs/fields.py
+++ b/wcs/fields.py
@@ -2226,6 +2226,14 @@ class ItemsField(WidgetField, ItemFieldMixin):
     raise ValueError('invalid data for items type (%r)' % value)

     def get_view_value(self, value, **kwargs):
+         # summary page
+         if 'value_id' in kwargs:
+             r = TemplateIO(html=True)
+             r += htmltext('<ul>')
+             for x in kwargs['value_id']:
+                 r += htmltext('<li>%s</li>' % x)
+             r += htmltext('</ul>')
+             return r.getvalue()
```

```
if type(value) is str: # == display_value
    return value
if value:
```

Constatation empirique que sur la page de résumé la méthode est appelée avec un `value_id` contrairement à la page de tableau. Mais dans `value` il y a déjà une chaîne de caractère "un, deux", tandis que dans `kwargs['value_id']` il y a une liste.

Ci-joint une capture, sans style associé il y a une marge supérieure et inférieure sur le `` qui n'est pas du plus bel effet.

(je veux bien abandonner ce ticket s'il est jugé plus fastidieux de me guider que de faire soi-même).

#5 - 25 février 2021 15:32 - Frédéric Péters

Constatation empirique que sur la page de résumé la méthode est appelée avec un `value_id` contrairement à la page de tableau.

Oui ce n'est pas nécessairement bien nommé "value_id", c'est nommé ainsi en pensant uniquement aux champs liste; ça vient de `get_value_info()` dans `wcs/fields.py`,

```
# return the selected value and an optional dictionary that will be
# passed to get_view_value() to provide additional details.
```

En pratique pour une liste à choix multiples ce qui se trouve mis dans "value_id" c'est la valeur "raw", la liste des choix.

J'écris "la liste des choix", sur un champ alimenté par une source de données, c'est plus précisément la liste des identifiants des choix, i.e.

```
+         for x in kwargs['value_id']:
+             r += htmltext('<li>%s</li>' % x)
```

va afficher 1 et 4, plutôt que "pomme" et "abricot".

Dans le `get_value_info` sont passées toutes les données disponibles, il y aurait dans le cas de la liste à choix multiples à en extraire les libellés individuels, qui peuvent se trouver, le stockage étant :

```
{ '1': ['1', '4'],
  '1_display': 'Pomme, Abricot',
  '1_structured': [{'id': 1, 'text': 'Pomme', 'nom': 'Pomme'}, {'id': 4, 'text': 'Abricot', 'nom': 'Abricot'}]
},
etc.
```

(il y a cependant à vérifier que la colonne `_structured` est bien enregistrée pour les sources de données qui contiennent juste id et text et pas d'attributs supplémentaires).

#6 - 02 mars 2021 18:41 - Emmanuel Cazenave

- Fichier `0001-fields-display-ItemsField-using-ul-on-summary-page-4.patch` ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposé changé de Non à Oui

Frédéric Péters a écrit :

(il y a cependant à vérifier que la colonne `_structured` est bien enregistrée pour les sources de données qui contiennent juste id et text et pas d'attributs supplémentaires).

Non ce n'est pas le cas, mais pas très gênant, ça se gère comme le cas sans data source.

Je tente un bout de css, il y aurait à faire la même chose côté `publik-base-theme` pour l'affichage front.

#7 - 02 mars 2021 19:23 - Frédéric Péters

Non ce n'est pas le cas, mais pas très gênant, ça se gère comme le cas sans data source.

C'est-à-dire que ça va afficher une liste d'identifiants plutôt que des libellés ?

#8 - 03 mars 2021 10:37 - Emmanuel Cazenave

Non ça va afficher les libellés.

La situation avant ce patch sur les paramètres reçus par `get_view_value` sur une page de résumé :

- pour une des choix rentrés manuellement ('foo', 'bar') ou venant d'une source de donnée "simple" (avec juste 'id' et 'text' : `{{'id': '1', 'text': 'foo'}, {'id': '2', 'text': 'bar'}}`) ça donne

```
value = 'foo, bar'  
value_id = ['foo', 'bar']
```

(avec aucune donnée en plus dispo via `get_structured_value` dans le cas de la source de donnée simple)

- pour une source de donnée "complexe" `{{'id': '1', 'text': 'foo', 'whatever': 'xxx'}, {'id': '2', 'text': 'bar', 'whatever': 'zzz'}}`, ça donne

```
value = 'foo, bar'  
value_id = ['1', '2']
```

mais avec des infos dispos en plus via `get_structured_value`.

Du coup le patche consiste juste à surcharger `get_value_info` pour avoir les libellés de dispo via un argument `text` dans le cas source de donnée complexe et d'aller piocher dans `text` ou `value_id` selon le cas.

#9 - 03 mars 2021 11:46 - Frédéric Péters

Ma question porte sur :

(il y a cependant à vérifier que la colonne `_structured` est bien enregistrée pour les sources de données qui contiennent juste `id` et `text` et pas d'attributs supplémentaires).

i.e. le cas où il y a une source de données, qui serait `{{'id': '1', 'text': 'foo'}, {'id': '2', 'text': 'bar'}}`.

#10 - 03 mars 2021 12:06 - Emmanuel Cazenave

Frédéric Péters a écrit :

Ma question porte sur :

(il y a cependant à vérifier que la colonne `_structured` est bien enregistrée pour les sources de données qui contiennent juste `id` et `text` et pas d'attributs supplémentaires).

La réponse est non, pas de colonne `_structured` enregistrée dans ce cas, mais dans ce cas toujours la liste des libellés est passée dans `value_id` (déjà le cas avant ce patch), donc j'ai juste à piocher dedans.

#11 - 03 mars 2021 12:15 - Emmanuel Cazenave

- Statut changé de *Solution proposée* à *En cours*

Oulala grosse confusion, la source de donnée sur laquelle je testais expose les mêmes données dans 'id' et 'text', désolé pour le bruit.

#12 - 03 mars 2021 16:22 - Emmanuel Cazenave

- Fichier `0001-fields-display-ItemsField-using-ul-on-summary-page-i.patch` ajouté

Du coup tentative d'avoir à disposition la colonne `_structured` pour le cas data source "simple" c'est :

```
-         if not set(structured_options[0].keys()) != set(['id', 'text']):  
-             return
```

J'observe que ça fait échouer des tests qui vérifient l'absence de cette donnée, ça semble facile corriger, mais zéro hauteur de vue sur cette situation, est-ce la voie à suivre ?

Pour le reste du patch ambitions revues à la baisse : afficher sous forme de liste uniquement si on a le `_structured` à disposition, sinon continuer avec l'affichage actuel (je ne vois pas de chemin pour assurer un affichage liste quelque soit la situation).

#13 - 03 mars 2021 16:41 - Frédéric Péters

J'observe que ça fait échouer des tests qui vérifient l'absence de cette donnée, ça semble facile corriger, mais zéro hauteur de vue sur cette situation, est-ce la voie à suivre ?

Je dirais que oui, ça me semble avoir été fait ainsi en pensant uniquement au type "liste" et pour économiser un peu de données. (origine de tout ça

dans #7466).

Pour le reste du patch ambitions revues à la baisse : afficher sous forme de liste uniquement si on a le `_structured` à disposition

C'est-à-dire "pas pour les anciennes demandes", right ?

#14 - 04 mars 2021 13:44 - Emmanuel Cazenave

- Fichier `0001-fields-store-_structured-on-ItemsField-for-every-dat.patch` ajouté
- Fichier `0002-fields-display-ItemsField-using-ul-on-summary-page-i.patch` ajouté
- Fichier `0003-css-remove-vertical-margin-on-dataview-list-44123.patch` ajouté
- Statut changé de *En cours* à *Solution proposée*

Frédéric Péters a écrit :

C'est-à-dire "pas pour les anciennes demandes", right ?

Oui tout à fait.

#15 - 16 mars 2021 10:48 - Frédéric Péters

Ok, sans doute unir 0002 & 0003 en un seul commit et ajouter une classes sur le `` pour ne pas cibler large.

#16 - 16 mars 2021 12:12 - Emmanuel Cazenave

- Fichier `0001-fields-store-_structured-on-ItemsField-for-every-dat.patch` ajouté
- Fichier `0002-fields-display-ItemsField-using-ul-on-summary-page-i.patch` ajouté

Frédéric Péters a écrit :

et ajouter une classes sur le `` pour ne pas cibler large.

J'étais partis là dessus puis j'ai vu que tout ça se passe dans un `div.field-type-items`, il m'a semblé opportun de s'en servir plutôt que d'introduire une nouvelle classe.

#17 - 16 mars 2021 13:43 - Frédéric Péters

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

Ok ainsi.

#18 - 16 mars 2021 14:12 - Emmanuel Cazenave

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

```
commit d44e6155ee9ca9323cd070ea25e1636833f31632
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date: Thu Mar 4 12:40:15 2021 +0100
```

```
fields: display ItemsField using <ul> on summary page if possible (#44123)
```

```
commit 149c73db4f3eba4a98ae1527ba5cd0b3308d8bdd
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date: Wed Mar 3 17:43:02 2021 +0100
```

```
fields: store _structured on ItemsField for every data source (#44123)
```

#19 - 16 mars 2021 16:16 - Frédéric Péters

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

#20 - 17 mars 2021 10:36 - Frédéric Péters

- Lié à *Development #52097*: *Styler les listes d'items à choix multiples* ajouté

#21 - 18 avril 2021 09:04 - Frédéric Péters

- Lié à *Development #50532*: *Le résumé du formulaire ne permet pas de représenter un champ "liste à choix multiple"* ajouté

Fichiers

Screenshot_2021-02-25 Backoffice de Démarches - choix multiple - 1.png	1,93 ko	25 février 2021	Emmanuel Cazenave
0001-fields-display-ItemsField-using-ul-on-summary-page-4.patch	2,97 ko	02 mars 2021	Emmanuel Cazenave
0001-fields-display-ItemsField-using-ul-on-summary-page-i.patch	3,09 ko	03 mars 2021	Emmanuel Cazenave
0001-fields-store-_structured-on-ItemsField-for-every-dat.patch	3,74 ko	04 mars 2021	Emmanuel Cazenave
0002-fields-display-ItemsField-using-ul-on-summary-page-i.patch	3,61 ko	04 mars 2021	Emmanuel Cazenave
0003-css-remove-vertical-margin-on-dataview-list-44123.patch	706 octets	04 mars 2021	Emmanuel Cazenave
0001-fields-store-_structured-on-ItemsField-for-every-dat.patch	3,74 ko	16 mars 2021	Emmanuel Cazenave
0002-fields-display-ItemsField-using-ul-on-summary-page-i.patch	4,07 ko	16 mars 2021	Emmanuel Cazenave