w.c.s. - Development #13166

accès à _structured dans les variables

12 septembre 2016 16:46 - Frédéric Péters

Statut:	Fermé	**Début:**	12 septembre 2016
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

Par sûr des conséquences :

```
diff --git a/wcs/formdata.py b/wcs/formdata.py
index 6743278..64da8fc 100644
--- a/wcs/formdata.py
+++ b/wcs/formdata.py
@@ -75,6 +75,7 @@ def get_dict_with_varnames(fields, data, formdata=None, varnames_only=False):
                     new_data['var_%s_raw' % field.varname] = raw_value
             if field.store_structured_value and data is not None:
                 structured_value = data.get('%s_structured' % field.id)
+                new_data['var_%s_structured' % field.varname] = structured_value
                 if type(structured_value) is dict:
                     for k, v in structured_value.items():
                         if k in ('id', 'text'):
```

Historique

#1 - 12 septembre 2016 16:49 - Frédéric Péters

Ça donne un truc utile comme ça : form_var_blah_structured = [{'text': 'pomme', 'price': '5', 'id': '1'}, {'text': 'poire', 'price': '2', 'id': '2'}] mais ça m'effraie un peu de sortir ça ainsi de la db, je préférerais plutôt pour cette situation avoir un form_var_blah_list_price = ['5', '2'], mais ça suppose que tous les dictionnaires ont les mêmes clés.

#2 - 12 septembre 2016 16:55 - Frédéric Péters

Dans cet esprit, en considérant que se baser sur les clés du premier élément est suffisant :

```
--- a/wcs/formdata.py
+++ b/wcs/formdata.py
@@ -86,6 +86,12 @@ def get_dict_with_varnames(fields, data, formdata=None, varnames_only=False):
                             if k in ('id', 'text'):
                                 continue
                             new_data['var_%s_%s_%s' % (field.varname, i, k)] = v
+                    if len(structured_value):
+                        for k in structured_value[0].keys():
+                            if k in ('id', 'text'):
+                                continue
+                            new_data['var_%s_list_%s' % (field.varname, k)] = [
+                                    x.get(k) for x in structured_value]
     return new_data

 def flatten_dict(d):
```

#3 - 12 septembre 2016 16:57 - Frédéric Péters

L'idée derrière c'est d'avoir une série de cases à cocher qui soit tirées d'une data source avec id/text + une clé "price"; et derrière de facilement faire le total. sum([x for x in form_var_blah_list_price if x]) voire sum(form_var_blah_list_price), ça me semble bien.

#4 - 12 septembre 2016 16:58 - Frédéric Péters

sum([Decimal(x) for x in form_var_blah_list_price if x])

#5 - 08 mai 2020 15:45 - Frédéric Péters

- Statut changé de Nouveau à Fermé

Il me semble que c'est bon avec #40203 puis #41855.