

w.c.s. - Bug #27247

Application de l'obligation de saisir un champs selon l'affichage de ce champs (condition d'affichage sur le champs)

11 octobre 2018 13:41 - Mikaël Ates (de retour le 29 avril)

Statut:	Fermé	Début:	11 octobre 2018
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		

Description

Je souhaite qu'un champs obligatoire ne le soit que s'il est affiché. Je rend ce champs obligatoire et je pose une condition d'affichage dessus.

Le fonctionnement est celui souhaité sauf lorsque je reviens sur la page contenant ce champs en cliquant sur précédent. Pour produire cela je passe sur la page sans avoir le champs affiché, je fais suivant et j'arrive sans alerte sur la page suivante. Je clique sur précédent. Je fais afficher le champs obligatoire et je ne le complète pas. Je clique sur suivant et je passe à la page suivante sans alerte. Je fais à nouveau précédent. Je ne modifie rien et clique sur suivant. Il y a cette fois ci une alerte sur le formulaire. Je modifie ensuite sur cette page un champs faisant désafficher le champs conditionnel. Je clique sur suivant. Une erreur de complétion est affiché en haut de formulaire. Je clique à nouveau sur suivant, je passe à la page suivante sans erreur affichée.

Cela se passe ici : <https://demarches-departement06.test.entrouvert.org/social/demande-apa/>

Le champs concerné est le suivant : <https://demarches-departement06.test.entrouvert.org/backoffice/forms/18/fields/129/>
Son affichage dépend de la réponse 'Oui' à la question posée dans le champs précédent.

Révisions associées

Révision 25a033b7 - 23 octobre 2018 10:29 - Frédéric Péters

forms: don't let conditional pages alter evaluation of field visibility (#27247)

Historique

#1 - 11 octobre 2018 13:42 - Mikaël Ates (de retour le 29 avril)

- Description mis à jour

#2 - 11 octobre 2018 13:54 - Frédéric Péters

Tu peux attacher un export du formulaire à ce ticket ? Éventuellement réduire aux seuls champs impliqués ? Éventuellement faire une vidéo du problème ?

Parce que j'ai essayé de suivre la description et reproduire sans ces éléments, sans succès, <https://perso.entrouvert.org/~fred/tmp/condition-obligatoire.ovg>

#3 - 11 octobre 2018 13:55 - Frédéric Péters

Vidéo ou produire une liste d'actions courtes amenant le résultat,

- choisir oui
- cliquer sur suivant
- cliquer sur précédent
- choisir non
- cliquer sur suivant
 - ça aurait du dire que le champ xxx était obligatoire.

#5 - 11 octobre 2018 15:37 - Frédéric Péters

- Assigné à mis à Frédéric Péters

```
def test_field_condition_on_required_field(pub):  
    # from https://dev.entrouvert.org/issues/27247  
    FormDef.wipe()  
    formdef = FormDef()
```

```

formdef.name = 'Foo'
formdef.fields = [
    fields.PageField(id='0', label='1st page', type='page'),
    fields.ItemField(type='item', id='1', label='Bar',
        items=['oui', 'non'],
        required=True, varname='bar'),
    fields.ItemField(type='item', id='2', label='Foo', size='40',
        required=True,
        items=['plop'],
        condition={'type': 'django', 'value': 'form_var_bar == "oui"'}),
    fields.PageField(id='3', label='1st page', type='page'),
    fields.CommentField(type='comment', id='4', label='HELLO!'),
]
formdef.store()

```

```

app = get_app(pub)
resp = app.get('/foo/')
assert 'f1' in resp.form.fields
assert 'f2' in resp.form.fields
assert resp.html.find('div', {'data-field-id': '1'}).attrs['data-live-source'] == 'true'
assert resp.html.find('div', {'data-field-id': '2'}).attrs.get('style') == 'display: none'
resp.form['f1'] = 'non'
live_resp = app.post('/foo/live', params=resp.form.submit_fields())
assert live_resp.json['result']['1']['visible']
assert not live_resp.json['result']['2']['visible']
resp = resp.form.submit('submit')
assert 'HELLO' in resp.body
resp = resp.form.submit('previous')
resp.form['f1'] = 'oui'
live_resp = app.post('/foo/live', params=resp.form.submit_fields())
assert live_resp.json['result']['1']['visible']
assert live_resp.json['result']['2']['visible']
resp = resp.form.submit('submit')
assert '<div class="error">required field</div>' in resp.body
assert 'HELLO' not in resp.body

```

#6 - 11 octobre 2018 15:42 - Frédéric Péters

Sauf que c'est là normal, parce que le <select> a un seul élément, forcément sélectionné.

L'exemple côté 06 est plutôt :

```

fields.ItemField(type='item', id='2', label='Foo', size='40',
    required=True,
    hint='---',
    items=['plop'],
    condition={'type': 'django', 'value': 'form_var_bar == "oui"'}),

```

et si je fais ça, le test fonctionne correctement, l'utilisateur est bien interrompu.

#7 - 11 octobre 2018 16:22 - Frédéric Péters

Trouvé ce qui enclenche le bug, c'est la présence d'une condition sur la page qui suit, j'ai donc maintenant un test unitaire qui échoue correctement.

```

fields.PageField(id='3', label='1st page', type='page',
    condition={'type': 'python', 'value': 'True'}),

```

#8 - 11 octobre 2018 17:18 - Frédéric Péters

Ce qui se passe c'est qu'à l'évaluation des pages (notamment pour l'affichage en barre latérale), il y a un ConditionVars() qui est ajouté au contexte, avec les données de la demande dans son état présent (i.e. avant traitement des champs de la page en cours).

Du coup plus loin quand ce sont les champs qui sont évalués pour voir s'ils sont visibles ou pas, ça tape dessus, ça considère le champ pas visible, et donc son critère "obligatoire" n'est pas pris en compte.

Tous les tests (de test_form_pages) passent, y compris le nouveau, avec cette modification, qui basiquement annule l'ajout du ConditionVars.

```

--- a/wcs/forms/root.py
+++ b/wcs/forms/root.py
@@ -522,6 +522,7 @@ class FormPage(Directory, FormTemplateMixin):
     if self._pages:
         return self._pages
     current_data = self.get_transient_formdata().data
+    sources = get_publisher().substitutions.sources[:]
     pages = []

```

```

        field_page = None
        for field in self.formdef.fields:
@@ -532,6 +533,7 @@ class FormPage(Directory, FormTemplateMixin):
        if not field_page: # form without page fields
            pages = [None]
            self._pages = pages
+
        get_publisher().substitutions.sources = sources
        return pages

def reset_pages_cache(self):

```

C'est plutôt moche, il y aurait bien sûr moyen de faire ça avec un with ...: mais ce serait sur le fond pareillement moche.

Une autre possibilité ça me semble être de construire le CompatibilityNamesDict en y conservant l'ordre des sources, pour qu'à l'évaluation, la source "transient&lazy formdata" qui est tapée en haut de la pile soit testé avant de tomber sur le ConditionVars (mais ça demande de remplacer le CompatibilityNamesDict(dict) par quelque chose vraisemblablement moins performant).

Aussi, il y aurait à voir comment tout simplement dégager ConditionVars(), voir dans quelle mesure le "transient formdata" pourrait être créé plus tôt et déjà utiliser pour l'évaluation des étapes. (mais quelques tentatives naïves ont amenés quelques tests en erreur).

Au final, ça n'est de toute façon pas pour ce soir, je vais donc dormir dessus et voir. (mais il y a des chances quand même pour finalement m'arrêter à la solution moche et reporter un travail de fond à plus tard).

#9 - 12 octobre 2018 09:33 - Frédéric Péters

- Fichier 0001-forms-don-t-let-conditional-pages-alter-evaluation-o.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Pas de grandes manœuvres mais quand même un contextmanager, plus propre/lisible.

#10 - 22 octobre 2018 16:35 - Thomas Noël

J'aurais pensé ce contextmanager dans un PageCondition::evaluate du genre :

```

class PageCondition(Condition):
    ...

    def evaluate(self):
        with get_publisher().substitutions.freeze():
            return super().evaluate()

```

... mauvaise idée ? ("sécurité" superflue ?)

#11 - 22 octobre 2018 16:41 - Frédéric Péters

Il est possible qu'à d'autres moments on profite de l'effet de bord, donc je suis frileux là.

#12 - 23 octobre 2018 10:15 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Frédéric Péters a écrit :

Il est possible qu'à d'autres moments on profite de l'effet de bord, donc je suis frileux là.

Ack donc ; le commentaire dans le code rappelle bien le contexte un peu spécial de l'affaire.

#13 - 23 octobre 2018 10:32 - Frédéric Péters

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

```

commit 25a033b72877c9f29cd20456db7c3e999e686cbc
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Fri Oct 12 09:15:52 2018 +0200

```

```

forms: don't let conditional pages alter evaluation of field visibility (#27247)

```

#14 - 23 décembre 2018 14:36 - Frédéric Péters

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

