

## w.c.s. - Development #35903

### Valeur choisie invalide dans la saisie d'un formulaire de workflow

07 septembre 2019 09:18 - Nicolas Roche

<b>Statut:</b>	Fermé	<b>Début:</b>	07 septembre 2019
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Frédéric Péters	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
<p>Affiche "valeur choisie invalide" à la soumission d'un formulaire de workflow (c'est à dire ajouté en tant qu'action d'un état dans un workflow) sur une liste remplie depuis une source de donnée JSON tout en utilisant en paramètre un champ issu du même formulaire.</p> <p>Tout est dit ci-dessus mais au risque de me répéter je préfère étayer un petit peu :</p> <p>Dans un formulaire crée depuis la fabrique de workflow on a un champ issu de l'URL JSON suivante : [passerelle_url]/csvdatasource/XXX/query/YYYY/?param=[form2_var_var]</p> <ul style="list-style-type: none"><li>• ok si var vient du formulaire FO</li><li>• ok si var vient d'une variable de traitement</li><li>• ok si var vient d'un autre formulaire BO</li><li>• ko var si vient du même formulaire BO</li></ul> <p>Le champ n'a pas besoin d'être obligatoire pour reproduire le bug. Le bug est reproductible en front-office et en back-office</p>			
<b>Demandes liées:</b>			
Lié à w.c.s. - Development #32960: évaluation dynamique des conditions pour l...		<b>Fermé</b>	<b>10 mai 2019</b>
Lié à w.c.s. - Bug #35363: pas de calcul live des listes dans un formulaire d...		<b>Fermé</b>	<b>12 août 2019</b>
Lié à w.c.s. - Bug #34564: Dans les formulaire de workflow, le contexte ne co...		<b>Fermé</b>	<b>05 juillet 2019</b>
Lié à w.c.s. - Development #38852: Évaluation de plusieurs listes live enchaî...		<b>Nouveau</b>	<b>08 janvier 2020</b>

#### Révisions associées

##### Révision e8384bec - 22 octobre 2019 11:13 - Nicolas Roche

tests: check for live data sources in workflow forms (#35903)

##### Révision 9da3ee99 - 22 octobre 2019 11:13 - Frédéric Péters

misc: recreate workflow form after submission, to get live items (#35903)

#### Historique

##### #2 - 09 septembre 2019 14:34 - Nicolas Roche

- Fichier menu.csv ajouté

- Fichier workflow-form-bo-within-datasource.wcs ajouté

En attendant d'avoir quelque chose de mieux borné, voici un workflow et un CSV qui mettent en évidence le bug.

Il faut configurer une requête 'choix' qui filtre le CSV sur l'id.

Le workflow peut être appelé depuis un formulaire vide.

##### #3 - 10 septembre 2019 17:53 - Nicolas Roche

Voici un test qui permet de reproduire l'erreur.

Il reprend test\_field\_live\_select\_content afin de tester [#27173](#) non plus sur le formulaire principale, mais sur un second formulaire affiché au cours du workflow.

```
def test_field_live_select_content_on_display_form(pub):
    create_user(pub)
    wf = Workflow(name='wf-title')
```

```

st1 = wf.add_status('Status1', 'st1')

data_from_data_source = [{'id': '1', 'text': 'un'}, {'id': '2', 'text': 'deux'}]

# form displayed into workflow
display_form = FormWorkflowStatusItem()
display_form.id = '_x'
display_form.by = ['_submitter']
display_form.varname = 'xxx'
display_form.formdef = WorkflowFormFieldsFormDef(item=display_form)
display_form.formdef.fields = [
    fields.StringField(type='string', id='1', label='Foo', varname='foo'),
    fields.ItemField(type='item', id='2', label='Bar', varname='bar',
                    data_source={
                        'type': 'json',
                        'value': "https://api.example.com/json?foo=[xxx_var_foo]"}),
]

st1.items.append(display_form)
display_form.parent = st1
wf.store()

# initial empty 'test' form
formdef = create_formdef()
formdef.fields = []
formdef.confirmation = False
formdef.workflow_id = wf.id
formdef.store()
formdef.data_class().wipe()

# submit initial empty form
app = get_app(pub)
resp = login(app, username='foo', password='foo').get('/test/')
resp = resp.form.submit('submit').follow()

# provide a value for 'f1' field
assert resp.html.find('div', {'data-field-id': '1'}).attrs['data-live-source'] == 'true'
assert resp.html.find('div', {'data-field-id': '2'}).find('select')
resp.form['f1'].value = 'foo'
live_resp = app.post('/test/1/live?modified_field_id=1', params=resp.form.submit_fields())
assert live_resp.json['result']['1']['visible']
assert live_resp.json['result']['2']['visible']

# simulate javascript filling the <select>
resp.form['f2'].options = []
for item in data_from_data_source:
    resp.form['f2'].options.append((item['id'], False, item['text']))
resp.form['f2'] = '2'

# submit the form
resp = resp.form.submit('submit') # <- hang here ...
# ... fails to retrieve the simulate javascript filled options for the select

resp = resp.follow()
assert 'The form has been recorded' in resp.body

```

Le test échoue dans `quixote/form/widget.py::SelectWidget:_parse_single_selection()` ...

```

for value, description, key in self.options:
    ...
else:
    if self.verify_selection:
        self.error = self.SELECTION_ERROR
        return default

```

... parce que les options (du select HTML) utilisées pour valider la valeur sont vides :

```

(Pdb) self.options
[(None, '---', '')]

```

#### #4 - 10 septembre 2019 17:55 - Nicolas Roche

- Sujet changé de Valeur choisie invalide dans la saisie en backoffice d'un formulaire à Valeur choisie invalide dans la saisie d'un formulaire de

**#5 - 10 septembre 2019 18:11 - Nicolas Roche**

- Description mis à jour

**#7 - 13 septembre 2019 17:48 - Stéphane Laget**

également ici : #35874

**#8 - 14 septembre 2019 21:17 - Benjamin Dauvergne**

Pour les ItemField le contenu de la datasource est mise en cache dès le premier appel à ItemField.get\_options() dans self.\_cached\_data\_source, je suppose un lien mais je vais lire le code de FormWorkflowStatusItem pour en être sûr...  
Pas le bon code.

Donc non, ce serait plutôt dans wcs/forms/common.py:FormStatusPage au niveau de get\_workflow\_form() :

```

def get_workflow_form(self, user):
    submitted_fields = []
    form = self.filled.get_workflow_form(user, displayed_fields=submitted_fields)
    if form:
        form.attrs['data-live-url'] = self.filled.get_url() + 'live'
    if form and form.is_submitted():
        with get_publisher().substitutions.temporary_feed(self.filled, force_mode='lazy'):
            # remove fields that could be required but are not visible
            self.filled.evaluate_live_workflow_form(user, form)
            get_publisher().substitutions.feed(self.filled)
            for field in submitted_fields:
                if not field.is_visible(self.filled.data, self.formdef) and 'f%s' % field.id in form._name
s:
                del form._names['f%s' % field.id]
    return form

```

on voit bien le feed des données actuelle du formulaire mais pas le feed des données qui viennent d'être soumises comme je pense on doit l'avoir lors de la soumission d'un formulaire "classique" avec les histoires de get\_transient\_formdata, comme là :

```

submitted_fields = []
transient_formdata = self.get_transient_formdata()
with get_publisher().substitutions.temporary_feed(
    transient_formdata, force_mode='lazy'):
    form = self.create_form(page=page,
        displayed_fields=submitted_fields,
        transient_formdata=transient_formdata)

```

Voilà je pense que c'est cerné, Nicolas je pense qu'il te faudra la science de Fred pour avancer plus loin.

**#9 - 14 septembre 2019 22:25 - Benjamin Dauvergne**

Mauvaise analyse, les données sont bien ajoutées aux substitutions par :

```
self.filled.evaluate_live_workflow_form(user, form)
```

mais comme on est dans le contexte d'un temporary\_feed c'est défaut, et donc quand on arrive là :

```

279 def check_submitted_form(self, form):
280     if form and form.is_submitted() and not form.has_errors():
281         url = self.submit(form)

```

le form.has\_errors() renvoie True puisqu'on a plus les valeurs, il faudrait ajouter le même code dans check\_submitted\_form, comme ceci :

```

def check_submitted_form(self, form):
    if form and form.is_submitted():
        with get_publisher().substitutions.temporary_feed(self.filled, force_mode='lazy'):
            # remove fields that could be required but are not visible
            self.filled.evaluate_live_workflow_form(user, form)
            get_publisher().substitutions.feed(self.filled)
            for field in submitted_fields:
                if not field.is_visible(self.filled.data, self.formdef) and 'f%s' % field.id in form._name
es:
                del form._names['f%s' % field.id]

    if not form.has_errors():
        url = self.submit(form)
    if url is None:

```

```
        url = get_request().get_frontoffice_url()
response = get_response()
response.set_status(303)
response.headers[str('location')] = url
response.content_type = 'text/plain'
return "Your browser should redirect you"
```

ou alors refactoriser un peu tout ce code la pour limiter la duplication.

#### #11 - 19 septembre 2019 11:30 - Nicolas Roche

- Fichier `workflow-form-wf-with-datasource.wcs` ajouté
- Fichier `menu.ods` ajouté

(correction)

Le WF et la source de donnée suivants cernent mieux le bug.

#### #12 - 19 septembre 2019 11:38 - Nicolas Roche

- Lié à [Development #32960](#): *évaluation dynamique des conditions pour les formulaires de workflow* ajouté

#### #13 - 19 septembre 2019 11:38 - Nicolas Roche

- Lié à [Bug #35363](#): *pas de calcul live des listes dans un formulaire de workflow* ajouté

#### #14 - 19 septembre 2019 11:38 - Nicolas Roche

- Lié à [Bug #34564](#): *Dans les formulaire de workflow, le contexte ne contient pas les variables locale (xxx\_var\_foo)* ajouté

#### #15 - 19 septembre 2019 11:59 - Nicolas Roche

- Fichier `0001-workflows-reevaluate-form-according-to-possible-new-.patch` ajouté
- Statut changé de *Nouveau* à *En cours*
- *Patch proposed* changé de *Non* à *Oui*

Merci Benjamin pour le coup de main.

Mauvaise analyse, les données sont bien ajoutées aux substitutions par :

```
self.filled.evaluate_live_workflow_form(user, form)
```

c'est justement là que ça plante.

Le bug est le suivant :

Le message "valeur choisie invalide" est enregistré par `quixote/form/widget.py::SelectWidget::_parse_single_selection()` parceque `formdata.workflow_data` contient encore la valeur du formulaire initial lors de l'évaluation (par `evaluate_live_workflow_form`) des champs qui sont soumis.

Donc (si j'ai bien compris) un problème qui se mort la queue mais qui a déjà été résolu dans l'autre point d'entrée 'live()' en réitérant l'appel.

Le patch corrige le test et propose une première correction (que je ne maîtrise pas mais que je vais essayer d'affiner) :

Suite au développement [#32960](#).

cette correction viens en fait compléter la correction [#35363](#).

en appliquant la même correction que celle appliquée dans 'live()' pour [#34564](#).

#### #16 - 19 septembre 2019 12:18 - Benjamin Dauvergne

Ma fois je n'y comprends rien du tout, t'as juste sorti le code dans le with, hors du with, je suppose que s'il y a un contexte il doit y avoir une raison.

#### #17 - 19 septembre 2019 17:15 - Nicolas Roche

- Fichier `0002-workflows-correct-test-and-variable-typo-35903.patch` ajouté
- Fichier `0003-workflows-rename-variable-and-move-a-line-to-ease-ne.patch` ajouté
- Fichier `0004-workflows-factorize-code-evaluating-workflow-form-35.patch` ajouté
- Fichier `0005-workflows-remove-redondant-code-WARNING-35903.patch` ajouté
- Statut changé de *En cours* à *Solution proposée*

Finalement, je n'ai rien trouvé de mieux.

Les patchs suivants corrigent 2 coquilles et factorisent le code.

t'as juste sorti le code dans le with, hors du with

Oui, du coup j'ai tenté une simplification dans le dernier patch proposé.  
Les tests passent, mais en fait j'ai l'impression que la correction apportée n'y est pour rien,  
et donc je vais déplacer ce dernier patch dans un autre ticket ([#36278](#)).

#### #18 - 19 septembre 2019 17:29 - Benjamin Dauvergne

Je vais passer la main pour la relecture, ça ne ressemble pas à ce que je pensais être la correction et je n'ai pas particulièrement d'idée sur ce qui serait juste. Juste renommer `submitted_fields` en `display_fields` me semble gratuit.

#### #19 - 19 septembre 2019 17:29 - Frédéric Péters

C'est cinq itérations sur un même patch, attache un seul patch.

#### #20 - 19 septembre 2019 17:32 - Frédéric Péters

Et les affaires sont suffisamment compliquées comme ça, comme écrit Benjamin, ne viens pas mêler des changements cosmétiques.

#### #21 - 19 septembre 2019 17:54 - Nicolas Roche

- Fichier `0001-workflows-simplify-cache-context-within-get_workflow.patch` ajouté

#### #22 - 19 septembre 2019 17:55 - Nicolas Roche

- Fichier `0001-workflows-simplify-cache-context-within-get_workflow.patch` supprimé

#### #23 - 19 septembre 2019 17:57 - Nicolas Roche

- Fichier `0001-workflows-reevaluate-form-according-to-possible-new-.patch` ajouté

oui, j'en conviens, désolé.  
(mauvais patch ajouté ci-dessus)

#### #24 - 19 septembre 2019 22:11 - Benjamin Dauvergne

Nicolas Roche a écrit :

oui, j'en conviens, désolé.  
(mauvais patch ajouté ci-dessus)

À mon avis le commentaire de Fred sur #36728 s'applique aussi :

Commentaire sur le patch vu dans la branche ça simplifie pas du tout ça modifie tout le comportement de w.c.s. qui suit le bout de code, sur des situations ultra-particulières qui ne se produisent sans doute pas dans les tests. (genre des combinaisons de pages conditionnelles, utilisant du python, avec des champs conditionnels et du django et des sources de données elles-mêmes utilisant l'un ou l'autre, etc.).

Dans le code suivant :

```
def get_workflow_form(self, user):
    form, submitted_fields = self.evaluate_workflow_form(user)
    if form:
        form.attrs['data-live-url'] = self.filled.get_url() + 'live'
    if form and form.is_submitted():
        with get_publisher().substitutions.temporary_feed(self.filled, force_mode='lazy'):
            # remove fields that could be required but are not visible
            self.filled.evaluate_live_workflow_form(user, form)
            get_publisher().substitutions.feed(self.filled)
            for field in submitted_fields:
                if not field.is_visible(self.filled.data, self.formdef) and 'f%s' % field.id in form._name
s:
                del form._names['f%s' % field.id]
    return form
```

le with protège le code qui vient après d'être perturbé par ces deux lignes qui modifie le contexte des variables de substitutions :

```
self.filled.evaluate_live_workflow_form(user, form)
get_publisher().substitutions.feed(self.filled)
```

tu ne peux juste pas faire :

```
+ self.filled.evaluate_live_workflow_form(user, form)
+ get_publisher().substitutions.unfeed(lambda x: x is self.filled)
+ get_publisher().substitutions.feed(self.filled)
```

en dehors du with, ça change complètement le contexte d'exécution.

Le problème n'est même pas là, il est sur ces lignes :

```
289 def check_submitted_form(self, form):
290     if form and form.is_submitted() and not form.has_errors():
```

c'est dans form.has\_errors() que quixote/form/widget.py::SelectWidget::\_parse\_single\_selection() dont tu parles plus haut est appelé, c'est autour de cette ligne qu'il faut changer des choses.

## #25 - 20 septembre 2019 09:24 - Frédéric Péters

En fait c'est vraiment super d'avoir le test pour reproduire le problème et ça va vraiment aider mais la correction en elle-même, parce qu'elle doit prendre en compte des dangers invisibles, je pense maintenant que ça doit être pour ma pomme.

## #26 - 20 septembre 2019 09:47 - Nicolas Roche

Merci Fred !

(Je regarde si je peux pas fournir un second test).

Au cas où un dernier petit laïus sur le sujet.

---8<---

Oui, j'ai bien has\_error() à True, mais je ne vois pas comment l'exploiter.

Voici la pile des appels :

```
...
/home/nroche/src/wcs/wcs/forms/common.py(242)_q_index()
-> form = self.get_workflow_form(user)
/home/nroche/src/wcs/wcs/forms/common.py(271)get_workflow_form()
-> self.filled.evaluate_live_workflow_form(user, form)
/home/nroche/src/wcs/wcs/formdata.py(576)evaluate_live_workflow_form()
-> wf_status.evaluate_live_form(form, self, user)
/home/nroche/src/wcs/wcs/workflows.py(1467)evaluate_live_form()
-> item.evaluate_live_form(form, filled, user)
/home/nroche/src/wcs/wcs/wf/form.py(199)evaluate_live_form()
-> self.formdef.fields, self.formdef.get_data(form),
/home/nroche/src/wcs/wcs/formdef.py(710)get_data()
-> d.update(self.get_field_data(field, widget))
/home/nroche/src/wcs/wcs/formdef.py(686)get_field_data()
-> d[field.id] = widget.parse()
/home/nroche/envs/publik-env/local/lib/python2.7/site-packages/quixote/form/widget.py(125)parse()
-> self._parse(request)
/home/nroche/envs/publik-env/local/lib/python2.7/site-packages/quixote/form/widget.py(466)_parse()
-> self.value = self._parse_single_selection(parsed_key)
/home/nroche/envs/publik-env/local/lib/python2.7/site-packages/quixote/form/widget.py(413)_parse_single_selection()
-> return default
```

Par exemple, dans get\_data j'ai tenté de positionner widget.verify\_selection à False sur les liste afin d'éviter l'erreur. Cela permet de valider le formulaire (de passer sans encombre dans check\_submitted\_form comme Benjamin le propose dans sa seconde analyse), mais ensuite on ne récupère pas les bonnes valeurs.

formdef.py:

```
def get_data(self, form):
    d = {}
    for field in self.fields:
        widget = form.get_widget('f%s' % field.id)
        if widget:
            d.update(self.get_field_data(field, widget))
    return d
```

Au début je pensais intervenir sur evaluate\_live\_form car c'est ici que l'on observe que la valeur du camps 1 (qui permet de récupérer les options du select du champ 2) est restée la même que celle fourni par le précédent formulaire validé (formulaire de démarche ou formulaire de workflow validé précédemment). Mais je n'ai pas réussi :

wf/form.py:

```
def evaluate_live_form(self, form, formdata, user):
++   print "avant: %s" % formdata.workflow_data
      workflow_data = {}
      for k, v in get_dict_with_varnames(
          self.formdef.fields, self.formdef.get_data(form),
          varnames_only=True).items():
          workflow_data['%s_%s' % (self.varname, k)] = v
      formdata.update_workflow_data(workflow_data)
++   print "apres: %s" % formdata.workflow_data
```

A vrai dire le test fourni ne reproduit pas cela (il me faudrait le rajouter à la fin du test `test_backoffice_workflow_form_with_live_data_source` de [#35363](#)). Le test fourni permet cependant de montrer que les variables de workflow sont mises à jour à ce moment.

avant: None

apres: {'xxx\_var\_bar2': 'plop', 'xxx\_var\_foo\_raw': None, 'xxx\_var\_foo': None, 'xxx\_var\_bar': 'hello'}

Or les options du select sont récupérées en utilisant la variable de workflow du champ 1 par la méthode `formdata.py::FormData::get_workflow_form()` appelée (**ici**) avant l'instruction qui met à jour cette variable (**là**) :

common.py:

```
def get_workflow_form(self, user):
    submitted_fields = []
    form = self.filled.get_workflow_form(user, displayed_fields=submitted_fields) # <- *ici*
    if form:
        form.attrs['data-live-url'] = self.filled.get_url() + 'live'
    if form and form.is_submitted():
        with get_publisher().substitutions.temporary_feed(self.filled, force_mode='lazy'):
            # remove fields that could be required but are not visible
            self.filled.evaluate_live_workflow_form(user, form) # <- *là*
    ...
```

Le test reproduit la même erreur ('valeur choisie invalide') parce que le champs 1 n'est pas encore valué dans `formdata` lors de la récupération des options du select.

Pour moi se serait plutôt à ce moment ~~ici~~ qu'il faudrait réévaluer les variables de workflow (ou inversement ~~là~~, ne pas les valider tant que l'on ne les a pas évaluées). Ce qui est fait par [#34564](#) (*reevaluate workflow form according to possible new content*) en faisant deux passes : la première passe est fautive (et l'erreur est ignorée) mais les 2 passes ont le mérite de rester cohérentes.

#### #27 - 20 septembre 2019 12:08 - Frédéric Péters

- Fichier `0001-tests-check-for-live-data-sources-in-workflow-forms-patch` ajouté

- Fichier `0002-misc-recreate-workflow-form-after-submission-to-get-patch` ajouté

0001 avec le test corrigé, pour garder l'attribution à Nicolas parce que c'était quand même le gros du travail.

0002 avec la correction, que je laisse découvrir.

#### #28 - 20 septembre 2019 12:42 - Nicolas Roche

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

arf, je m'en veux d'être passé à côté.

Ça passe les tests sur mon workflow.

(en fait le test de [#35363](#) ne permettrait pas plus de tester avec un champ 1 déjà initialisé)

#### #29 - 20 septembre 2019 12:47 - Benjamin Dauvergne

Y a ça qui est perdu au passage quand on recrée le form il me semble (surtout en cas d'erreur de validation, si la soumission passe c'est moins grave):

```
if form:
    form.attrs['data-live-url'] = self.filled.get_url() + 'live'
```

à déplacer en sortie de fonction je pense.

#### #30 - 20 septembre 2019 14:14 - Nicolas Roche

idem, il y a aussi

```
submitted_fields = []
```

apporté par le patch misc-don-t-check-displayed-fields-twice-in-live-eval.patch de [#35363](#) qui a sauté.

### #31 - 20 septembre 2019 16:27 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

### #32 - 20 septembre 2019 16:27 - Benjamin Dauvergne

- Fichier 0004-tox-remove-skipdist-set-scgi-max-version.patch ajouté
- Fichier 0005-misc-evaluate-workflow-form-three-times-to-support-d.patch ajouté
- Fichier 0002-misc-recreate-workflow-form-after-submission-to-get-.patch ajouté
- Fichier 0001-tests-check-for-live-data-sources-in-workflow-forms-.patch ajouté
- Fichier 0003-tests-check-longer-dependency-chain-in-workflow-form.patch ajouté
- Tracker changé de Bug à Development
- Statut changé de Solution validée à Solution proposée

Bon le problème semble plus compliqué, j'ai ajouté un deuxième ItemField 'foo2' dépendant du premier 'foo', créant ainsi une chaîne de dépendance de longueur 3, et ça ne marche plus. Il semble qu'il faille évaluer les formulaire n+1 fois si on a au maximum des chaînes de dépendance de longueur n (et il faudrait prendre en compte les conditions de visibilité). De fait ça ressemble à la recherche d'un point fixe dans l'évaluation d'un dataflow (= une feuille de tableur avec des formules). Pour couvrir tous les cas il faudrait un code commun entre l'évaluation live et normale qui ferait en gros ceci en pseudo-python (et je me demande comment c'est géré dans le code des formulaires normaux) :

```
form = get_form()

if submitted:
    while True:
        old_form = form
        invalidate_cache()
        evaluate_live_form(form)
        form = get_form()
        remove_invisible_fields(form)
        if compare(form, old_form) == 0:
            break
```

L'important ce serait de définir cette fonction compare pour savoir quand on peut s'arrêter.

Il y a une possibilité que ça boucle infiniment (genre foo1.visible = 'not foo2'; foo2.visible = 'not foo1..') si on a une dépendance cyclique, donc faudrait se limiter genre à un maximum de n+1 évaluations et expliquer dans la doc qu'on peut pas avoir des dépendances sur plus de n champs.

PS: je note ici pour mon édification personnelle que

```
get_publisher().substitutions.invalidate_cache()
```

et

```
get_publisher().substitutions.unfeed(lambda x: x is filled)
```

donne le même résultat (j'ai testé les deux pour voir dans live/get\_workflow\_form, pas bien regardé ce que ça change en vrai)

### #33 - 20 septembre 2019 16:40 - Frédéric Péters

Sérieux, on peut ne pas tous travailler sur la même chose ?

### #34 - 20 septembre 2019 16:41 - Benjamin Dauvergne

Frédéric Péters a écrit :

Sérieux, on peut ne pas tous travailler sur la même chose ?

Je signale juste que la solution apportée ne marche pas et je me demande si ça marche en front et comment; je trouve bien qu'on soit plusieurs à comprendre, ma branche n'apporte pas la solution, elle montre les problèmes qui restent.

### #35 - 20 septembre 2019 17:08 - Frédéric Péters

- Statut changé de Solution proposée à En cours
- Assigné à changé de Benjamin Dauvergne à Frédéric Péters



ma branche n'apporte pas la solution

Je modifie le ticket pour refléter ça, et confirmer que je m'occupe de ce ticket, merci.

**#36 - 20 septembre 2019 17:17 - Benjamin Dauvergne**

Frédéric Péters a écrit :

ma branche n'apporte pas la solution

Je modifie le ticket pour refléter ça, et confirmer que je m'occupe de ce ticket, merci.

Ah oui pardon, j'ai fait un git sub j'ai oublié de dire non à la question de prendre le ticket, pardon, j'ai supprimé ma branche aussi pour ne pas polluer.

**#37 - 04 octobre 2019 11:16 - Benjamin Dauvergne**

Le client demande si ce sera corrigé d'ici fin octobre.

**#38 - 04 octobre 2019 12:03 - Frédéric Péters**

Si le cas précis correspond au patch, ça peut aller, modulo modification pour reprendre l'attribut data-live-url; ça dépend de l'acceptation d'un patch qui se limiterait à ça. S'il s'agit d'un changement plus large, je n'ai pas regardé.

**#39 - 11 octobre 2019 17:23 - Frédéric Péters**

- Statut changé de *En cours* à *Information nécessaire*

**#40 - 11 octobre 2019 17:43 - Benjamin Dauvergne**

- Statut changé de *Information nécessaire* à *Solution validée*

Ok go pour le patch ça corrige le souci immédiat de Toulouse; pour le reste tant pis, on redécouvrira le problème plus tard quand quelqu'un mettra 3 ItemField enchaînés au lieu de 2.

**#41 - 11 octobre 2019 17:53 - Frédéric Péters**

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

Il y a quand même encore le data-live-url à assurer.

**#42 - 11 octobre 2019 18:02 - Benjamin Dauvergne**

De mon côté je vais ouvrir un autre ticket et rebaser ma branche avec le problème à long terme, j'essayerai de voir si ça affecte le front aussi.

**#44 - 21 octobre 2019 15:06 - Frédéric Péters**

- Fichier *0001-tests-check-for-live-data-sources-in-workflow-forms-.patch* ajouté

- Fichier *0002-misc-recreate-workflow-form-after-submission-to-get-.patch* ajouté

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

Voilà la version qui doit assurer le taf pour le cas rencontré.

**#45 - 22 octobre 2019 11:07 - Thomas Noël**

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

**#46 - 22 octobre 2019 11:14 - Frédéric Péters**

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

```
commit 9da3ee99014ff61bd636a63d83bd7b8cfea3213b
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Fri Sep 20 12:00:14 2019 +0200
```

```
misc: recreate workflow form after submission, to get live items (#35903)
```

```
commit e8384becebfa7f75a102b0474699a4354110711c
```

Author: Nicolas ROCHE <nroche@entrouvert.com>

Date: Tue Sep 10 10:43:22 2019 +0200

tests: check for live data sources in workflow forms (#35903)

#### #47 - 22 octobre 2019 12:15 - Frédéric Péters

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

#### #48 - 08 janvier 2020 22:33 - Benjamin Dauvergne

- Lié à Development #38852: Évaluation de plusieurs listes live enchaînées des les formulaires ajouté

#### Fichiers

---

menu.csv	91 octets	09 septembre 2019	Nicolas Roche
workflow-form-bo-within-datasource.wcs	2,22 ko	09 septembre 2019	Nicolas Roche
workflow-form-wf-with-datasource.wcs	1,77 ko	19 septembre 2019	Nicolas Roche
menu.ods	8,98 ko	19 septembre 2019	Nicolas Roche
0001-workflows-reevaluate-form-according-to-possible-new-.patch	4,81 ko	19 septembre 2019	Nicolas Roche
0002-workflows-correct-test-and-variable-typo-35903.patch	1,56 ko	19 septembre 2019	Nicolas Roche
0003-workflows-rename-variable-and-move-a-line-to-ease-ne.patch	2,5 ko	19 septembre 2019	Nicolas Roche
0004-workflows-factorize-code-evaluating-workflow-form-35.patch	2,28 ko	19 septembre 2019	Nicolas Roche
0005-workflows-remove-redondant-code-WARNING-35903.patch	1,54 ko	19 septembre 2019	Nicolas Roche
0001-workflows-reevaluate-form-according-to-possible-new-.patch	6,02 ko	19 septembre 2019	Nicolas Roche
0001-tests-check-for-live-data-sources-in-workflow-forms-.patch	3,86 ko	20 septembre 2019	Frédéric Péters
0002-misc-recreate-workflow-form-after-submission-to-get-.patch	1,29 ko	20 septembre 2019	Frédéric Péters
0004-tox-remove-skipdist-set-scgi-max-version.patch	790 octets	20 septembre 2019	Benjamin Dauvergne
0005-misc-evaluate-workflow-form-three-times-to-support-d.patch	2,96 ko	20 septembre 2019	Benjamin Dauvergne
0002-misc-recreate-workflow-form-after-submission-to-get-.patch	1,29 ko	20 septembre 2019	Benjamin Dauvergne
0001-tests-check-for-live-data-sources-in-workflow-forms-.patch	3,86 ko	20 septembre 2019	Benjamin Dauvergne
0003-tests-check-longer-dependency-chain-in-workflow-form.patch	3,55 ko	20 septembre 2019	Benjamin Dauvergne
0001-tests-check-for-live-data-sources-in-workflow-forms-.patch	4,3 ko	21 octobre 2019	Frédéric Péters
0002-misc-recreate-workflow-form-after-submission-to-get-.patch	1,72 ko	21 octobre 2019	Frédéric Péters