w.c.s. - Development #56824

Ne pas perdre les données _display et _structured des champs Item*Field quand la source de donnée renvoie None à la soumission

09 septembre 2021 12:59 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	09 septembre 2021
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Dans wcs/forms/root.py:FormPage.submitted() on a :

```
    filled.data = self.formdef.get_data(form)
```

Là selon qu'on est en mode page de validation, form contient les données sérialisées via create_view_form(form_data) ou form_data est extrait purement du draft ou bien un mélange des données qui viennent d'être soumises dans la dernière page et des données actuellement dans le draft (

```
            form_data = session.get_by_magictoken(magictoken, {})
            data = self.formdef.get_data(form)
            form_data.update(data)
```

)

```
    @classmethod
    def get_field_data(cls, field, widget):
        d = {}
        d[field.id] = widget.parse()
        if d.get(field.id) is not None and field.convert_value_from_str:
            d[field.id] = field.convert_value_from_str(d[field.id])
        field.set_value(d, d[field.id])
        if getattr(widget, 'cleanup', None):
            widget.cleanup()
        return d

    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
```

où Field.set_value(d, d[field.id]) (surchargé uniquement par MapField)

```
    def set_value(self, data, value):
        data['%s' % self.id] = value
        if self.store_display_value:
            display_value = self.store_display_value(data, self.id)
            if display_value:
                data['%s_display' % self.id] = display_value
            elif '%s_display' % self.id in data:
                del data['%s_display' % self.id]
        if self.store_structured_value and value:
            structured_value = self.store_structured_value(data, self.id)
            if structured_value:
                if isinstance(structured_value, dict) and structured_value.get('id'):
                    # in case of list field, override id
```

```
                    data['%s' % self.id] = str(structured_value.get('id'))
                data['%s_structured' % self.id] = structured_value
            elif '%s_structured' % self.id in data:
                del data['%s_structured' % self.id]
        elif self.store_structured_value and '%s_structured' % self.id in data:
            del data['%s_structured' % self.id]
```

La combinaison de tout ça c'est qu'on ignore complètement les valeurs existantes de _display, _structured qu'on possède déjà et que si la source foire à ce moment là on écrase avec None. Ou bien même si les valeurs diffèrent des valeurs vues par l'usager, on écrase les valeurs validées par lui avec des nouvelles.

Je pense que le formdata.data d'origine devrait être passé à toutes ces méthodes pour conserver les valeurs existantes pour _display et _structured s'il y en a.

J'ai l'impression que le patch attaché ferait l'affaire (il ne s'intéresse qu'à ce chemin particulier dans le code, soumission front et back); je ne trouve pas d'autre endroit ou formdef.get_data(..) soit utilisé pour écrasé complètement formdata.data avec des données régénérées de formdata.data lui même.

Je laisse le soin à quelqu'un qui a plus le temps que moi de voir ce qu'il en est.

Révisions associées

Révision 0e9658f3 - 30 novembre 2021 15:37 - Benjamin Dauvergne

formdef: make get_data raise on error if necessary (#56824)

Révision 2542247c - 30 novembre 2021 15:37 - Benjamin Dauvergne

forms: redisplay page when datasource is unavailable (#56824)

Historique

#3 - 08 novembre 2021 10:14 - Frédéric Péters

> Je pense que le formdata.data d'origine devrait être passé à toutes ces méthodes pour conserver les valeurs existantes pour _display et _structured s'il y en a.

Je ne suis pas sûr du tout de ça; je serais plutôt à dire que s'il y a erreur temporaire d'appel le seul truc sûr c'est lever une exception et bloquer l'avancée vers la page suivante.

#4 - 08 novembre 2021 12:32 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

Ok. Facile à dire mais on a pas envie que les chemins autour des DS raise de partout.

Est-ce que ça irait de lever dans Field.store_structured_value/Field.store_display/structured_value via un paramètre raise=False et uniquement si NameDataSource.get_display/structured_value retournent None ? On ajouterait ensuite un raise=False à FormDef.get_field_data/get_data pour obtenir une exception là où FormDef.get_data est utilisé.

Ensuite il faut que je fasse le tour de tous les formdef.get_data() pour voir ceux qui méritent ce traitement et il y en a pas mal à vérifier :

```
wcs/admin/users.py:96:                data = formdef.get_data(form)
wcs/backoffice/submission.py:340:        filled.data = self.formdef.get_data(form)
wcs/formdef.py:583:        data = self.workflow.variables_formdef.get_data(form)
wcs/forms/root.py:546:                transient_formdata.data.update(self.formdef.get_data(form))
wcs/forms/root.py:1064:                data = self.formdef.get_data(form)
wcs/forms/root.py:1085:                data = self.formdef.get_data(form)
wcs/forms/root.py:1123:                data = self.formdef.get_data(form)
wcs/forms/root.py:1192:            data = self.formdef.get_data(form)
wcs/forms/root.py:1341:        data = self.formdef.get_data(form)
wcs/forms/root.py:1434:        formdata.data.update(self.formdef.get_data(form))
wcs/forms/root.py:1449:        filled.data = self.formdef.get_data(form)
wcs/forms/root.py:1499:        new_data = self.formdef.get_data(form)
wcs/qommon/ident/idp.py:199:            data = formdef.get_data(form)
wcs/qommon/ident/password.py:565:            data = formdef.get_data(form)
wcs/qommon/ident/password.py:588:            data = formdef.get_data(form)
wcs/qommon/myspace.py:124:        data = formdef.get_data(form)
wcs/wf/form.py:238:            self.formdef.fields, self.formdef.get_data(form), varnames_only=True
```

#5 - 08 novembre 2021 12:53 - Frédéric Péters

D'un coup d'œil, ceux-ci concernent le formdef représentant les attributs de l'usager, on n'aura pas de source de données là-dedans, (et de toute façon pas d'édition depuis w.c.s.),

```
wcs/admin/users.py:96:                data = formdef.get_data(form)
wcs/qommon/ident/idp.py:199:            data = formdef.get_data(form)
wcs/qommon/ident/password.py:565:            data = formdef.get_data(form)
wcs/qommon/ident/password.py:588:            data = formdef.get_data(form)
wcs/qommon/myspace.py:124:        data = formdef.get_data(form)
```

côté action de workfllow j'aurais dit de toute façon rien à changer il y a d'autres choses à faire avant à ce sujet (#50795) mais à ouvrir le fichier c'est de toute façon juste là pour l'évaluation "live" du formulaire,

```
wcs/wf/form.py:238:            self.formdef.fields, self.formdef.get_data(form), varnames_only=True
```

Et celui-ci c'est dans le paramétrage options/variables de workflow, là sur une erreur c'est sans doute bien d'interrompre mais ce n'est pas non plus une situation fréquente, ça pourrait être décalé à un autre ticket,

```
wcs/formdef.py:583:        data = self.workflow.variables_formdef.get_data(form)
```

#6 - 09 novembre 2021 10:12 - Benjamin Dauvergne

J'ai commencé à faire de le tour des plus importants donc, ceux là :

```
wcs/forms/root.py:546:                transient_formdata.data.update(self.formdef.get_data(form))
wcs/forms/root.py:1064:                data = self.formdef.get_data(form)
wcs/forms/root.py:1085:                data = self.formdef.get_data(form)
wcs/forms/root.py:1123:                data = self.formdef.get_data(form)
wcs/forms/root.py:1192:            data = self.formdef.get_data(form)
wcs/forms/root.py:1341:        data = self.formdef.get_data(form)
wcs/forms/root.py:1434:        formdata.data.update(self.formdef.get_data(form))
wcs/forms/root.py:1449:        filled.data = self.formdef.get_data(form)
wcs/forms/root.py:1499:        new_data = self.formdef.get_data(form)
```

Ça me donne du code comme ça :

```
+            try:
+                return self.submitted(form, existing_formdata)
+            except SetValueError:
+                get_session().message = (
+                    'error',
+                    _('There was a temporary error during submission, please try again.'),
+                )
+                return redirect(filled.get_url())
```

Ici au niveau de submitted parce que le .get_data() est fait dedans et je préférais tout garder au niveau de FormPage._q_index. Mon souci c'est la dernière ligne, je ne sais pas trop où rediriger ou quoi afficher (via self.page()). Pour l'instant je renvoie un peu systématiquement vers filled.get_url() (dont je ne suis pas sûr non plus pour une soumission BO), ce qui va marcher à la soumission puisque filled est sensé contenir déjà les bonnes données, mais sur une transition entre pages je ne pense pas que ça fasse ce qu'on veut, réafficher la page en cours avec les données soumises. Est-ce qu'il y aurait une recette uniforme que tu me conseillerais ? J'ai posé un commentaire numéroté autour de chaque get_data non traité.

(Je pousse la branche wip avec tout ça).

#7 - 09 novembre 2021 10:30 - Frédéric Péters

Je n'ai pas regardé le patch et les différents cas, mais si c'est en cours de aisie, je pensais plutôt faire form.set_error(), proche de ce qu'on a dans le traitement des conditions de sortie de page,

```
@@ -1098,6 +1098,10 @@ class FormPage(Directory, FormTemplateMixin):
                        form.add(HiddenErrorWidget, 'post_condition%d' % i)
                        form.set_error('post_condition%d' % i, 'error')
                        page_error_messages.append(error_message)
+                if not page_error_messages and whatever_set_value_error:
+                    form.add(HiddenErrorWidget, 'transient_set_value_error')
+                    form.set_error('transient_set_value_error', 'error')
+                    page_error_messages.append('technical problem, try again')
```

et si c'est après la page de validation, quelque chose du même genre aussi, avec sans doute des chose à ajouter pour l'affichage de l'erreur, parce que je pense qu'on n'a pas ça pour cette page,

```
             magictoken = form.get_widget('magictoken').parse()
             form_data = session.get_by_magictoken(magictoken, {})
-            data = self.formdef.get_data(form)
+            try:
```

```
+                 data = self.formdef.get_data(form)
+             except SetValueError:
+                 form.add(HiddenErrorWidget, 'transient_set_value_error')
+                 form.set_error('transient_set_value_error', 'error')
+                 page_error_messages.append('technical problem, try again')
+                 return self.validating(form_data)
+
              form_data.update(data)
              session.add_magictoken(magictoken, form_data)
```

De manière générale, pas get_session().message + redirect.

#8 - 09 novembre 2021 23:23 - Benjamin Dauvergne

- Fichier 0001-formdef-make-get_data-raise-on-error-if-necessary-56.patch ajouté

- Tracker changé de Bug à Development

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

- Fichier 0002-forms-redisplay-page-when-datasource-is-unavailable-.patch ajouté

Les tests couvrent presque tout sauf :

- le cas de l'action d'édition (edit_mode),
- le cas step == 1, je pense que ce code est mort en fait, car à aucun endroit on émet de formulaire avec add_hidden('step', '1') ou alors ça devrait être -1.

Les parametrize/range c'est pour tenter de couvrir tous les cas, c'est un peu crado mais pas pire que le code que ça couvre :/

#10 - 10 novembre 2021 09:03 - Frédéric Péters

> le cas step 1, je pense que ce code est mort en fait

On passe bien dans ce code : https://jenkins.entrouvert.org/job/wcs/3120/cobertura/forms/root_py/#1180

(j'écris cela sans regarder les patchs mais c'est le seul endroit avec step 1), on y arrive via dix lignes plus haut step = 1 # so it will flow to submit.

Ou tu veux dire que ce code on passe dedans mais il n'apporte en fait rien ?

#11 - 10 novembre 2021 09:57 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

Oui je me suis un peu emballé, c'est le cas sans page de confirmation, je vais couvrir ça.

#12 - 10 novembre 2021 10:28 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Ou tu veux dire que ce code on passe dedans mais il n'apporte en fait rien ?

En voulant couvrir la gestion d'un SetValueError, effectivement je vois que ce bout de code marqué avec des étoiles ne fait rien visiblement, vu que juste avant lorsqu'on initialise step = 1 on recrée un formulaire quixote vide de champs du formulaire w.c.s. :

```
                    step = 1  # so it will flow to submit
                    # kind of restore state
                    form = Form()
                    form.add_hidden('step', '-1')
                    form.add_hidden('page', '-1')
                    form.add_hidden('magictoken', '-1')
                    form.add_submit('cancel')
                    if self.has_draft_support():
                        form.add_submit('removedraft')
                        form.add_submit('savedraft')
...
        if step == 1:
            form.add_submit('previous')
            magictoken = form.get_widget('magictoken').parse()
```

```
            if form.get_submit() == 'previous':
                return self.previous_page(len(self.pages), magictoken)
            magictoken = form.get_widget('magictoken').parse()
            form_data = session.get_by_magictoken(magictoken, {})
            data = self.formdef.get_data(form) ****
            form_data.update(data) ****
```

Je vais tenter de virer ce qui ne semble servir à rien.

#13 - 10 novembre 2021 12:15 - Benjamin Dauvergne

- Fichier 0001-forms-redisplay-page-when-datasource-is-unavailable-.patch ajouté

- Statut changé de En cours à Solution proposée

#14 - 11 novembre 2021 09:58 - Benjamin Dauvergne

- Fichier 0002-forms-redisplay-page-when-datasource-is-unavailable-.patch ajouté

- Fichier 0001-formdef-make-get_data-raise-on-error-if-necessary-56.patch ajouté

Avec le premier patch manquant.

#15 - 12 novembre 2021 15:05 - Frédéric Péters

Il reste un

```
+                # XXX: Should we handle datasources error here ?
```

faut-il le regarder, ou ça a été fait et juste oublié d'être retiré ?

(je serais à penser qu'il doit être géré également)

#16 - 12 novembre 2021 16:06 - Benjamin Dauvergne

- Fichier 0002-forms-redisplay-page-when-datasource-is-unavailable-.patch ajouté

- Fichier 0001-formdef-make-get_data-raise-on-error-if-necessary-56.patch ajouté

J'ai enlevé le commentaire, je ne sais pas comment le traiter proprement: on est déjà arrivé sur la page suivante, c'est le calcul du transient_formdata qui alimentera les variables de substitution pour le calcul des expressions de pré-remplissage; je ne vois pas comment m'arrêter à cet endroit, il faudrait revenir à la page précédente ou alors simplement afficher un bouton 'erreur, recharger la page.

#17 - 23 novembre 2021 12:01 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Benjamin Dauvergne a écrit :

> J'ai enlevé le commentaire, je ne sais pas comment le traiter proprement: on est déjà arrivé sur la page suivante, c'est le calcul du transient_formdata qui alimentera les variables de substitution pour le calcul des expressions de pré-remplissage; je ne vois pas comment m'arrêter à cet endroit, il faudrait revenir à la page précédente ou alors simplement afficher un bouton 'erreur, recharger la page.

Le transient_formdata initialisé ici contient les données mises à jour après pré-remplissage pour être utilisées ensuite dans les calculs des sources données pour des champs qui en dépendent; ex.:

```
page n:
field1 = ItemField(data_source=ds1_json, pré-rempli)
field2 = ItemField(data_source=ds2_json qui dépend de field1)
```

Si on arrive sur la page "n", que field1 se pré-remplit mais qu'on arrive pas à extraire les valeurs de la source de donnée ds1_json, le calcul de ds2_json pourrait être faussé et donc le champ vide ou avec les mauvais items, mais ça ne devrait pas passer un autosave ou la prochaine soumission donc ça n'a pas d'importance, les form_data en session ou dans le draft ne sont pas impactées puisque le pré-remplissage ne touche qu'à chaque widget.value et request.form.

#18 - 23 novembre 2021 12:03 - Frédéric Péters

- Statut changé de Solution validée à Solution proposée

#19 - 30 novembre 2021 09:58 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

Passons ça maintenant. (je tagguerai dans la journée une fois poussé).

#20 - 30 novembre 2021 15:38 - Frédéric Péters

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

poussés moi-même,

```
commit 2542247c27a708f12ec8342ca89a2fc75fefae2c
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Tue Nov 9 12:28:26 2021 +0100

    forms: redisplay page when datasource is unavailable (#56824)

commit 0e9658f3d0691c7394db1a26eeeb8961b963e178
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Tue Nov 9 09:52:38 2021 +0100

    formdef: make get_data raise on error if necessary (#56824)
```

#21 - 30 novembre 2021 17:17 - Frédéric Péters

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

Fichiers

fix-set_value.diff	4,05 ko	09 septembre 2021	Benjamin Dauvergne
0002-forms-redisplay-page-when-datasource-is-unavailable-.patch	16,8 ko	09 novembre 2021	Benjamin Dauvergne
0001-formdef-make-get_data-raise-on-error-if-necessary-56.patch	7,93 ko	09 novembre 2021	Benjamin Dauvergne
0001-forms-redisplay-page-when-datasource-is-unavailable-.patch	19,7 ko	10 novembre 2021	Benjamin Dauvergne
0002-forms-redisplay-page-when-datasource-is-unavailable-.patch	19,7 ko	11 novembre 2021	Benjamin Dauvergne
0001-formdef-make-get_data-raise-on-error-if-necessary-56.patch	7,93 ko	11 novembre 2021	Benjamin Dauvergne
0002-forms-redisplay-page-when-datasource-is-unavailable-.patch	19,1 ko	12 novembre 2021	Benjamin Dauvergne
0001-formdef-make-get_data-raise-on-error-if-necessary-56.patch	7,93 ko	12 novembre 2021	Benjamin Dauvergne