

w.c.s. - Development #33186

Initialisation d'un brouillon

17 mai 2019 10:10 - Brice Mallet

Statut:	Fermé	Début:	17 mai 2019
Priorité:	Normal	Echéance:	17 janvier 2020
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Développer une action de WF qui créerait un brouillon dans le cours du WF : En tant qu'utilisateur :			
<ul style="list-style-type: none">• action interactive (affichage d'un bouton)• dans le cours du WF, il ne serait pas possible de choisir la démarche concernée par l'initialisation du nouveau brouillon			
En tant qu'administrateur fonctionnel :			
<ul style="list-style-type: none">• paramétrage de cette action avec choix de la démarche du nouveau brouillon (identique au formulaire initial et liste explicite)• une fois choix du formulaire d'arrivée effectué, pouvoir régler, champ par champ, si un champ fait l'objet d'une recopie, ou pas			
NB : toute ressemblance avec une action existante serait purement fortuite.			
Demandes liées:			
Lié à w.c.s. - Development #39656: création d'une demande, option pour repren...		Fermé	07 février 2020
Lié à w.c.s. - Development #39637: création d'une demande, ne pas proposer le...		Fermé	07 février 2020
Lié à w.c.s. - Development #39638: création d'une demande ou création d'un br...		Fermé	07 février 2020
Lié à w.c.s. - Development #39639: création d'une demande, option pour affich...		Fermé	07 février 2020
Bloqué par w.c.s. - Development #39356: faire en sorte que FormData.get_url r...		Fermé	28 janvier 2020

Révisions associées

Révision 5388fda4 - 07 février 2020 11:46 - Benjamin Dauvergne

formdef: ease access to widget fields (#33186)

Révision 9bbe4b48 - 07 février 2020 11:46 - Benjamin Dauvergne

formdata: ease iteration of evolutions parts (#33186)

Révision 7f421d15 - 07 février 2020 11:46 - Benjamin Dauvergne

add new action create-formdata (#33186)

Historique

#1 - 17 mai 2019 10:11 - Brice Mallet

- Description mis à jour

#2 - 17 mai 2019 11:07 - Benjamin Dauvergne

Brice Mallet a écrit :

- action interactive (affichage d'un bouton)

En général on ne fait plus d'action qui affiche un boutons, on utilise les sauts par généricité.

#3 - 17 mai 2019 11:18 - Thomas Noël

Benjamin Dauvergne a écrit :

Brice Mallet a écrit :

- action interactive (affichage d'un bouton)

En général on ne fait plus d'action qui affiche un boutons, on utilise les sauts par généralité.

Oui, ça permettra d'utiliser l'action dans d'autres situations, par exemple de créer plusieurs demandes, etc. L'action "création d'un brouillon" devra créer ce qu'il faut dans les variables pour permettre l'affichage d'un lien vers le brouillon créé, ou la redirection directe de l'usager vers ce brouillon, etc.

choix de la démarche du nouveau brouillon (identique au formulaire initial et liste explicite)

C'est à dire, soit on coche "même formulaire" (permettant de généraliser le workflow), soit on coche

champ par champ, si un champ fait l'objet d'une recopie, ou pas

Plus généralement, pour un champ du formulaire cible, dire la valeur à lui attribuer, parmi : gabarit / expression Python / recopie du champ ayant le même identifiant.

Note : tout cela rend l'export/import de workflow assez hasardeux, il faudra sans doute rendre l'import intelligent (ne pas se baser que sur les slugs, prendre aussi en charge les noms des formulaires, etc)

#4 - 17 mai 2019 11:43 - Benjamin Dauvergne

Thomas Noël a écrit :

Benjamin Dauvergne a écrit :

Brice Mallet a écrit :

- action interactive (affichage d'un bouton)

En général on ne fait plus d'action qui affiche un boutons, on utilise les sauts par généralité.

Oui, ça permettra d'utiliser l'action dans d'autres situations, par exemple de créer plusieurs demandes, etc. L'action "création d'un brouillon" devra créer ce qu'il faut dans les variables pour permettre l'affichage d'un lien vers le brouillon créé, ou la redirection directe de l'usager vers ce brouillon, etc.

Plusieurs choses :

- ça m'irait qu'on ait un evolutionpart spécifique pour ce job, dans le futur ça pourrait permettre des interactions entre formulaires (ça peut nous aider à résoudre le souci de workflow parallèles, on lance d'autres workflow et après on pourrait attendre qu'ils atteignent un certain statut)
- le nouvel evolutionpart pourra exporter un LazyFormData pointant sur le nouveau formulaire (prendre en compte le cas où le brouillon est supprimé)
- ça pourrait être utile de ne pas se limiter au statut draft mais de pouvoir directement valider le formulaire (en lien avec mon premier point)

choix de la démarche du nouveau brouillon (identique au formulaire initial et liste explicite)

C'est à dire, soit on coche "même formulaire" (permettant de généraliser le workflow), soit on coche

champ par champ, si un champ fait l'objet d'une recopie, ou pas

Plus généralement, pour un champ du formulaire cible, dire la valeur à lui attribuer, parmi : gabarit / expression Python / recopie du champ ayant le même identifiant.

Oui.

Note : tout cela rend l'export/import de workflow assez hasardeux, il faudra sans doute rendre l'import intelligent (ne pas se baser que sur les slugs, prendre aussi en charge les noms des formulaires, etc)

Oui et prévoir une étape de recollement comme pour les statuts lors de l'import de workflo

#5 - 17 mai 2019 12:22 - Brice Mallet

"dans le futur ça pourrait permettre des interactions entre formulaires (ça peut nous aider à résoudre le souci de workflow parallèles, on lance d'autres workflow et après on pourrait attendre qu'ils atteignent un certain statut)

OUAAAAH, ce serait bien en effet

- ça pourrait être utile de ne pas se limiter au statut draft mais de pouvoir directement valider le formulaire (en lien avec mon premier point)

oui

Qui c'est qui chiffre tout cela ? ;-)

#6 - 18 mai 2019 14:29 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#7 - 18 mai 2019 14:29 - Benjamin Dauvergne

- Fichier 0004-formdata-ease-iteration-of-evolutions-parts-33186.patch ajouté

- Fichier 0005-add-new-action-create-formdata-33186.patch ajouté

- Fichier 0003-workflows-ease-filtering-common-fields-of-related-fo.patch ajouté

- Fichier 0001-formdef-ease-access-to-widget-fields-33186.patch ajouté

- Fichier 0002-workflows-ease-selecting-related-formdefs-33186.patch ajouté

- Tracker changé de Support à Development

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Proposition entamé à partir de ma vision des choses (pouvoir créer des brouillons ou soumettre, donner accès aux formulaires créer depuis le workflow d'origine).

Les 4 premiers patchs ajoutent des méthodes pour faciliter l'implémentation.

Il reste plusieurs choses déterminer fermement :

- choisir les noms des choses visible :
 - nom de l'action, ici 'Créer formulaire'
 - nom de la variable de substitution donnant accès aux nouveaux formulaires, ici sub_formdatas
 - choisir quoi faire au niveau du contexte de soumission :
 - j'ai repris la façon qu'a resoumission de faire un lien entre le formulaire d'origine et le nouveau, à voir si on veut garder ça, le rendre optionnel ou imaginer quelque chose de différent
 - je reprends l'utilisateur d'origine et le canal de soumission d'origine, idem à voir si on veut ça ou le rendre optionnel ou programmable
 - j'affiche un message dans l'historique (implémentation de view() sur mon EvolutionPart), pour être plus générique on ne devrait rien afficher et s'en remettre à l'action d'ajout d'un message dans l'historique
- du CSS à faire pour afficher correctement FieldOrExpressionWidget (ça ne s'affiche pas en ligne)

C'est testé localement en jouant dans w.c.s. mais donc il manque les tests (dans test_admin_pages.py pour le backoffice fonctionnel, dans test_workflows.py pour tester unitairement la machinerie et les variables de substitution, dans test_backoffice_pages et test_form_pages pour tester le fonctionnement complet).

Je ne compte pas finir, je laisse ça pour celui qui finira.

#8 - 18 mai 2019 14:41 - Frédéric Péters

- Statut changé de Solution proposée à Nouveau

- Assigné à Benjamin Dauvergne supprimé

- Patch proposed changé de Oui à Non

Je ne compte pas finir, je laisse ça pour celui qui finira.

Je corrige le ticket pour noter que tu ne proposes pas de patch.

Reste à une personne motivée à répondre à la question de Brice, "Qui c'est qui chiffre tout cela ?", à celle-ci de chiffrer, puis à développer les tickets w.c.s. qui seront associés.

#9 - 18 mai 2019 15:30 - Benjamin Dauvergne

Frédéric Péters a écrit :

Reste à une personne motivée à répondre à la question de Brice, "Qui c'est qui chiffre tout cela ?", à celle-ci de chiffrer, puis à développer les tickets w.c.s. qui seront associés.

Là ça m'a pris environ 5h entre le train hier et un peu aujourd'hui, je dirai encore 2j pour rentrer dans le code des tests et écrire les tests pour obtenir une couverture équivalente à resubmit. Je veux bien ouvrir le ticket et y accrocher mon code, plusieurs me semblent inutiles.

À noter dans mon code il manque aussi l'option même formulaire (ce n'est pas vraiment évident comparé à resubmit car il faudrait se restreindre aux champs communs entre les formulaires liés à ce workflow, et les repérer par leur varname, ici je les repère par leur id, c'est moins pérenne mais ça permet de viser plus de champs, c'est aussi un point discutable).

#14 - 04 septembre 2019 15:45 - Brice Mallet

- *Echéance mis à 02 mars 2020*

- *Assigné à mis à Emmanuel Cazenave*

#16 - 30 septembre 2019 10:49 - Benjamin Dauvergne

- *Assigné à changé de Emmanuel Cazenave à Benjamin Dauvergne*

Je reprends le ticket si on me donne juste le choix entre voir mon travail jeter à la poubelle ou tout faire moi même.

#20 - 12 octobre 2019 10:19 - Benjamin Dauvergne

- *Projet changé de Publik à w.c.s.*

#21 - 24 octobre 2019 15:45 - Benjamin Dauvergne

- *Echéance changé de 02 mars 2020 à 15 novembre 2019*

#22 - 09 novembre 2019 17:51 - Benjamin Dauvergne

- *Fichier 0003-add-new-action-create-formdata-33186.patch ajouté*

- *Fichier 0002-formdata-ease-iteration-of-evolutions-parts-33186.patch ajouté*

- *Fichier 0001-formdef-ease-access-to-fields-with-varname-33186.patch ajouté*

- *Statut changé de Nouveau à Solution proposée*

- *Patch proposed changé de Non à Oui*

J'ai retiré ce qui était devenu inutile, les tests arrivent, l'export XML est implémenté et tout se base sur les varname.

#23 - 25 novembre 2019 10:19 - Benjamin Dauvergne

- *Statut changé de Solution proposée à En cours*

#24 - 27 novembre 2019 23:58 - Benjamin Dauvergne

- *Fichier 0003-add-new-action-create-formdata-33186.patch ajouté*

- *Fichier 0005-add-first-test.patch ajouté*

- *Fichier 0002-formdata-ease-iteration-of-evolutions-parts-33186.patch ajouté*

- *Fichier 0004-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch ajouté*

- *Fichier 0001-formdef-ease-access-to-fields-with-varname-33186.patch ajouté*

Petit souci avec le mode pickle-lazy, bizarrement ça ne foire qu'en python3, je n'ai pas compris pourquoi.

PS: ça foire aussi en py2, tout va bien.

#25 - 28 novembre 2019 01:55 - Benjamin Dauvergne

- *Fichier 0003-add-new-action-create-formdata-33186.patch ajouté*

- *Fichier 0005-add-first-test.patch ajouté*

- *Fichier 0002-formdata-ease-iteration-of-evolutions-parts-33186.patch ajouté*

- *Fichier 0004-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch ajouté*

- *Fichier 0001-formdef-ease-access-to-fields-with-varname-33186.patch ajouté*

C'est pas évident à trouver del lazy['linked'].

#26 - 05 décembre 2019 13:14 - Brice Mallet

- Echéance changé de 15 novembre 2019 à 17 janvier 2020

#28 - 23 janvier 2020 10:25 - Benjamin Dauvergne

- Fichier 0003-add-new-action-create-formdata-33186.patch ajouté
- Fichier 0007-second-test.patch ajouté
- Fichier 0006-storage-handle-tuple-likes-in-deep_bytes2str.patch ajouté
- Fichier 0005-add-first-test.patch ajouté
- Fichier 0002-formdata-ease-iteration-of-evolutions-parts-33186.patch ajouté
- Fichier 0004-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch ajouté
- Fichier 0001-formdef-ease-access-to-fields-with-varname-33186.patch ajouté
- Statut changé de En cours à Solution proposée

Deuxième test, cette fois en soumission backoffice, je regarde un poil la couverture mais ça devrait être bon.

#29 - 23 janvier 2020 10:55 - Benjamin Dauvergne

- Fichier 0004-add-new-action-create-formdata-33186.patch ajouté
- Fichier 0007-second-test.patch ajouté
- Fichier 0006-add-first-test.patch ajouté
- Fichier 0001-storage-handle-tuple-likes-in-deep_bytes2str.patch ajouté
- Fichier 0003-formdata-ease-iteration-of-evolutions-parts-33186.patch ajouté
- Fichier 0005-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch ajouté
- Fichier 0002-formdef-ease-access-to-fields-with-varname-33186.patch ajouté

corrections python3 (toujours embêtant c'est machin.id qui sont des fois des entiers des fois des chaînes).

#30 - 23 janvier 2020 11:48 - Benjamin Dauvergne

- Fichier 0004-add-new-action-create-formdata-33186.patch ajouté
- Fichier 0007-second-test.patch ajouté
- Fichier 0006-add-first-test.patch ajouté
- Fichier 0001-storage-handle-tuple-likes-in-deep_bytes2str.patch ajouté
- Fichier 0003-formdata-ease-iteration-of-evolutions-parts-33186.patch ajouté
- Fichier 0005-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch ajouté
- Fichier 0002-formdef-ease-access-to-fields-with-varname-33186.patch ajouté

quelques modifications pour arriver à faire que deep_bytes2str() fonctionnent avec des tuples et des namedtuple.

#31 - 23 janvier 2020 12:39 - Benjamin Dauvergne

- Fichier 0004-add-new-action-create-formdata-33186.patch ajouté
- Fichier 0007-second-test.patch ajouté
- Fichier 0006-add-first-test.patch ajouté
- Fichier 0001-storage-handle-tuple-likes-in-deep_bytes2str.patch ajouté
- Fichier 0003-formdata-ease-iteration-of-evolutions-parts-33186.patch ajouté
- Fichier 0005-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch ajouté
- Fichier 0002-formdef-ease-access-to-fields-with-varname-33186.patch ajouté

FormDef.wipe() est mon ami.

#32 - 23 janvier 2020 14:34 - Benjamin Dauvergne

- Fichier 0004-add-new-action-create-formdata-33186.patch ajouté
- Fichier 0007-second-test.patch ajouté
- Fichier 0006-add-first-test.patch ajouté
- Fichier 0001-storage-handle-tuple-likes-in-deep_bytes2str.patch ajouté
- Fichier 0009-create_formdata-fix-XML-export-import-of-mappings-wi.patch ajouté
- Fichier 0003-formdata-ease-iteration-of-evolutions-parts-33186.patch ajouté
- Fichier 0005-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch ajouté
- Fichier 0010-tests-add-create_formdata-test-on-XML-export-import-.patch ajouté
- Fichier 0008-workflows-add-WorkflowStatusItem.__eq__-operator-331.patch ajouté
- Fichier 0002-formdef-ease-access-to-fields-with-varname-33186.patch ajouté

plus de tests.

#33 - 23 janvier 2020 15:00 - Frédéric Péters

```
+ # lick on first available formdata
```

c'est dégueulasse.

#34 - 23 janvier 2020 15:06 - Frédéric Péters

Par ailleurs, tu aurais quelques explications pour guider la relecture, que ça ne prenne pas des semaines ?

Et en attendant, pour revenir sur un détail de forme, "formdata" n'est pas un mot qu'on veut exposer dans l'interface.

```
+     if 'varname' in parameters:
+         form.add(VarnameWidget, '%svarname' % prefix,
+                 title=_('Identifiant'), value=self.varname,
+                 hint=_('This is used to get generated document in expressions.'),
+                 advanced=not (self.varname))
```

et on ne veut sans doute pas parler de document ici.

#35 - 23 janvier 2020 15:42 - Benjamin Dauvergne

- Fichier 0004-add-new-action-create-formdata-33186.patch ajouté
- Fichier 0007-second-test.patch ajouté
- Fichier 0006-add-first-test.patch ajouté
- Fichier 0001-storage-handle-tuple-likes-in-deep_bytes2str.patch ajouté
- Fichier 0009-create_formdata-fix-XML-export-import-of-mappings-wi.patch ajouté
- Fichier 0003-formdata-ease-iteration-of-evolutions-parts-33186.patch ajouté
- Fichier 0005-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch ajouté
- Fichier 0011-tests-add-create_formdata-admin-page-tests-33186.patch ajouté
- Fichier 0010-tests-add-create_formdata-test-on-XML-export-import-.patch ajouté
- Fichier 0008-workflows-add-WorkflowStatusItem.__eq__-operator-331.patch ajouté
- Fichier 0002-formdef-ease-access-to-fields-with-varname-33186.patch ajouté

Encore des tests.

#36 - 23 janvier 2020 15:49 - Benjamin Dauvergne

Frédéric Péters a écrit :

Et en attendant, pour revenir sur un détail de forme, "formdata" n'est pas un mot qu'on veut exposer dans l'interface.

Ok je remplace par form dans les deux labels.

[...]

et on ne veut sans doute pas parler de document ici.

Ok j'ai remplacé par :

```
- hint=_('This is used to get generated document in expressions. '),
+ hint=_('This is used to get linked forms in expressions. '),
```

branche à jour.

#37 - 23 janvier 2020 15:56 - Benjamin Dauvergne

Frédéric Péters a écrit :

Par ailleurs, tu aurais quelques explications pour guider la relecture, que ça ne prenne pas des semaines ?

Le code est assez proche de resubmit, les ajouts c'est principalement l'EvolutionPart qui permet de garder une référence vers les nouvelles demandes via la variable proxy linked et les différentes options (keep_user, backoffice_submission, draft) alors que dans resubmit tout est en dur ou presque.

Au niveau des tests:

- l'usage en front est testé en mode anonyme, donc resoumission et ajout du tracking code à la session pour permettre à l'utilisateur de naviguer vers la nouvelle demande,
- l'usage en backoffice avec conservation de l'utilisateur d'origine et création d'un draft en soumission backoffice,
- test import/export XML (au passage j'ajoute une méthode StatusItem.__eq__ pour tester qu'on ne perd rien à l'import)
- test sur la page d'administration pour voir si le widget d'édition des mappings fonctionne à minima en conservant les valeurs déjà présentes

Deux parties ne sont pas tellement testées:

- les proxy pour les variables de substitution,
- l'usage sans passer par un draft

#38 - 23 janvier 2020 15:57 - Frédéric Péters

Plus généralement je pense qu'attention doit être portée ici par Serghei, parce que l'action de création de fiche ne peut être qu'une évolution de l'action présentée ici.

#39 - 23 janvier 2020 15:59 - Frédéric Péters

Deux parties ne sont pas tellement testées:

Elles pourraient l'être ? Notamment il y a juste un {{ linked.url }} alors que LinkedFormdataSubstitutionProxy par sa taille donne l'impression de permettre davantage.

#40 - 23 janvier 2020 16:03 - Benjamin Dauvergne

Frédéric Péters a écrit :

Plus généralement je pense qu'attention doit être portée ici par Serghei, parce que l'action de création de fiche ne peut être qu'une évolution de l'action présentée ici.

En remplaçant les référence explicite à FormDef par un self.formdef_class et en cachant certains paramètre ça ne devrait pas trop poser de problème; il faudra aussi rendre customisable le nom de la variable proxy linked.

#41 - 23 janvier 2020 16:04 - Serghei Mihai

Frédéric Péters a écrit :

Plus généralement je pense qu'attention doit être portée ici par Serghei, parce que l'action de création de fiche ne peut être qu'une évolution de l'action présentée ici.

Tout à fait. On se disait cela à l'instant ici avec Thomas. L'interface de configuration de l'action mériterait un générique. Je relis plus en détails.

#42 - 23 janvier 2020 16:12 - Serghei Mihai

Dans:

```
def _fields_to_options(self, formdef):
    return [(None, '---', '')] + [
```

```
(field.varname, field.label, field.varname) for field in formdef.get_all_varname_fields()]
```

`get_all_varname_fields()` élimine les champs sans `varname`, et en plus expose les champs `backoffice` or tous les champs peuvent ne pas avoir de nom de variable.

Et les champs `backoffice` ne doivent pas forcément être remplis. Plutôt utiliser `formdef.get_widget_fields()`, non?

#43 - 23 janvier 2020 17:15 - Benjamin Dauvergne

Frédéric Péters a écrit :

Deux parties ne sont pas tellement testées:

Elles pourraient l'être ? Notamment il y a juste un `{{ linked.url }}` alors que `LinkedFormdataSubstitutionProxy` par sa taille donne l'impression de permettre davantage.

Voilà j'ai complété les tests:

- objet proxy testé plus complètement
- cas sans draft testé, ça a d'ailleurs levé un bug sur les variables de substitutions qui m'a obligé à entourer ça d'un `with get_publisher().substitutions.freeze()`
- ajout d'un champ `ItemField` pour tester la copie des données structurées

#44 - 23 janvier 2020 17:19 - Benjamin Dauvergne

Serghei Mihai a écrit :

Dans:

[...]

`get_all_varname_fields()` élimine les champs sans `varname`, et en plus expose les champs `backoffice` or tous les champs peuvent ne pas avoir de nom de variable.

Oui c'est voulu, on en a discuté à l'eocamp on ne peut pas référencer les champs de manière réutilisable sans passer par les `varname` ça, on a aucune action où on référence des champs de formulaire autrement que par leur `varname`.

Et les champs `backoffice` ne doivent pas forcément être remplis.

Je n'ai pas compris, sur une soumission direct on pourra remplir les champs BO sur un draft non (il n'y a pas de contrôle contre ça mais je sais que ça ne marchera pas). Il n'y aucune obligation de les remplir.

#45 - 23 janvier 2020 18:00 - Benjamin Dauvergne

Voilà, couverture à 91 %. Il reste une méthode `repr`, des `except` ou des cas limites.

https://jenkins.entrouvert.org/job/wcs-wip/job/wip%252F33186-Initialisation-d-un-brouillon/13/Coverage_20Report_20_28native_29/

#46 - 23 janvier 2020 18:54 - Benjamin Dauvergne

Dans la branche deux modifications pour Serghei :

- abstraction de `FormDef` et `LinkedFormdataEvolutionPart`
- utilisation de `field.id` pour les champs cibles au lieu de `field.varname` et uniquement les champs du formulaire, les champs BO sont désormais exclus

PS: j'ai poussé une branche consolidée au cas où <http://git.entrouvert.org/wcs.git/log/?h=wip/33186-Initialisation-d-un-brouillon-consolide>

#47 - 24 janvier 2020 07:41 - Frédéric Péters

Sur la forme toujours, parce que ça va continuer à inutilement m'ennuyer, je préférerais qu'on n'utilise pas de double préfixe `__`, que ça reste réservé à Python. (de manière "amusante" il y a une seule occurrence aujourd'hui, c'est `__key_prefix`, parce que code copié/collé/adapté depuis hobo).

#48 - 24 janvier 2020 10:54 - Benjamin Dauvergne

Frédéric Péters a écrit :

Sur la forme toujours, parce que ça va continuer à inutilement m'ennuyer, je préférerais qu'on n'utilise pas de double préfixe `__`, que ça reste réservé à Python. (de manière "amusante" il y a une seule occurrence aujourd'hui, c'est `__key_prefix`, parce que code copié/collé/adapté depuis hobo).

Ok, branche(s) à jour.

#49 - 28 janvier 2020 15:01 - Frédéric Péters

(je continue à faire des petites remarques mais j'aimerais une vraie relecture par quelqu'un d'autre)

~

Ce "linked" continue à m'ennuyer, il est désormais testé dans davantage de recoins mais à quoi donc va servir tout ça ?

Aussi, plutôt qu'entrer ainsi avec une variable à la racine, je trouve que ça aurait davantage sa place accédé depuis la demande, genre `form_links_<created>_url (<created> étant ici l'identifiant donné à l'action)`.

(pareillement les fichiers attachés auraient plutôt leur place en `form_attachments_...` et sont à la racine par incident historique).

```
--- a/wcs/qommon/storage.py
+++ b/wcs/qommon/storage.py
@@ -122,6 +122,10 @@ def deep_bytes2str(obj, seen=None):
...

```

Ça ne sert plus à rien de toucher à ça, on est en python 3.

À ce sujet sans doute des trucs à retirer aussi côté `force_text`, `force_str`.

~

`evolutionpart_class` → `evolution_part_class`; et je comprends que ces indirections viennent peut-être après mes commentaires à Serghei sur faire la relecture en pensant aux fiches mais autrement j'aurais juste dit d'éliminer cette indirection.

#50 - 28 janvier 2020 15:53 - Benjamin Dauvergne

Frédéric Péters a écrit :

Ce "linked" continue à m'ennuyer, il est désormais testé dans davantage de recoins mais à quoi donc va servir tout ça ?

L'idée c'est de pouvoir suivre les demandes créées depuis la demande d'origine: le cas d'usage le plus ancien que je connais c'est le fait de demander plusieurs validations pour une même demande, mais ça pourrait être aussi quand un truc est géré par deux services : un service accueil et un service technique avec des demandes de formes différentes, le service technique n'ayant pas forcément à connaître le nom du demandeur ou demandant plus de détails ou de mieux structurer une demande générique. Le même `EvolutionPart` pourrait servir à lier des demandes entre elles aussi (pour éviter les bidouilles +1), ça mais ça demanderait une action particulière.

Aussi, plutôt qu'entrer ainsi avec une variable à la racine, je trouve que ça aurait davantage sa place accédé depuis la demande, genre `form_links_<created>_url (<created> étant ici l'identifiant donné à l'action)`.

Ok, faut que je vois comment faire ça, mais j'ai l'impression qu'avec le mode lazy en fait `item.get_substitution_variables()` ne sert plus à rien si tout passe par `LazyFormData`. Si je comprends bien on ne doit plus jamais créer de variable du style `form_x_y_z` on doit tout poser dans `LazyFormData` et c'est `CompatibilityNamesDict` qui s'occupera des deux syntaxes ?

(pareillement les fichiers attachés auraient plutôt leur place en `form_attachments_...` et sont à la racine par incident historique).

Ok donc on devrait déprécier `WorkflowStatusItem.get_substitution_variables()` à terme.

Ça ne sert plus à rien de toucher à ça, on est en python 3.

Les tests ne passaient pas sans ça, mais le code a pu changer depuis.

À ce sujet sans doute des trucs à retirer aussi côté `force_text`, `force_str`.

Ok.

`evolutionpart_class` → `evolution_part_class`; et je comprends que ces indirections viennent peut-être après mes commentaires à Serghei sur faire la relecture en pensant aux fiches mais autrement j'aurais juste dit d'éliminer cette indirection.

Ok.

#51 - 28 janvier 2020 16:21 - Frédéric Péters

L'idée c'est de pouvoir suivre les demandes créées (...)

je comprends ça mais il y a différentes variations dans les appels, qui n'ont peut-être pas toutes une utilité, au moins dans des formes condensées,

- `linked.length`, nombre de fois où il y a eu une demande créée à partir de celle-ci, utile ?
- `linked.status`, statut de la demande liée, sans se soucier de l'identifiant de l'action
- `linked.resubmitted`, la même chose, pour l'action qui a "resubmitted" comme identifiant
- `linked.resubmitted.0.status`, la même chose mais si jamais il y avait plusieurs demandes créées à partir de l'action, prendre la première
- `linked.1.status`, la même chose que `linked.status` mais en prenant la deuxième demande créée
- `linked.var.foo_string`, un champ de la première demande liée.

On pourrait j'espère simplifier, en partant par exemple `form_links_`, avoir `form_links_resubmitted` (identifiant de l'action obligatoire), et derrière l'objet lié, et donc comme documentation on noterait que si tu veux `form_var_string` de la demande liée tu fais `form_links_resubmitted.form_var_links`.

Et je laisserais à plus tard la réflexion pour accommoder la situation avec plusieurs demandes créées depuis la même action. (dans l'immédiat ma position serait de continuer sur la lancée de `simili-queryset` de `LazyFormDefObjectsManager`, et avoir `form_links_resubmitted_set`, et donc `form_links_resubmitted_set.pending` pour toutes les demandes créées par l'action et encore en attente, etc.).

~~

avec le mode lazy (...) `CompatibilityNamesDict`

L'idée est bien d'arriver à l'utiliser de manière systématique, avec `CompatibilityNamesDict` pour gérer la syntaxe en underscores.

~~

Les tests ne passaient pas sans ça, mais le code a pu changer depuis.

Normalement on ne devrait passer dans `deep_bytes2str` que les objets pickle anciens, et comme tout ça sera fait avant que cette action ne soit employée, jamais rencontrer ici de tuple ou namedtuple.

#52 - 28 janvier 2020 16:33 - Serghei Mihai

Les méthodes `get_varname_fields`, `get_all_widget_fields`, `get_all_varname_fields` ne servent pas (ou plus), autant ne pas les déclarer.

#53 - 28 janvier 2020 17:21 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Ok donc on devrait déprécier `WorkflowStatusItem.get_substitution_variables()` à terme.

Ça ne sert plus à rien de toucher à ça, on est en python 3.

Les tests ne passaient pas sans ça, mais le code a pu changer depuis.

Bon finalement ça servait à quelque chose :

```
[2020-01-28 16:03:15] exception caught
Exception:
  type = '<class 'AttributeError'>', value = '<'Mapping' object has no attribute '__dict__'>'

Stack trace (most recent call first):
  File "/var/lib/jenkins/workspace/86-Initialisation-d-un-brouillon/wcs/qommon/storage.py", line 134, in deep_bytes2str
    132     seen[id(obj)] = True
    133     if hasattr(obj, '__class__') and obj.__class__.__module__.startswith(('wcs.', 'qommon.', 'modules.'
)):
> 134         obj.__dict__ = deep_bytes2str(obj.__dict__, seen)
    135         return obj
    136     return obj
```

Je vais rajouter un `not isinstance(obj, tuple)` et un commentaire.

#54 - 28 janvier 2020 17:28 - Frédéric Péters

Faudrait idéalement voir pourquoi ça arrive là-dedans, parce que normalement ça ne devrait pas. (mais sinon, oui passons et je chercherai moi-même plus tard).

#55 - 28 janvier 2020 17:33 - Benjamin Dauvergne

Frédéric Péters a écrit :

L'idée c'est de pouvoir suivre les demandes créées (...)

je comprends ça mais il y a différentes variations dans les appels, qui n'ont peut-être pas toutes une utilité, au moins dans des formes condensées,

- `linked.length`, nombre de fois où il y a eu une demande créée à partir de celle-ci, utile ?
- `linked.status`, statut de la demande liée, sans se soucier de l'identifiant de l'action
- `linked.resubmitted.status`, la même chose, pour l'action qui a "resubmitted" comme identifiant
- `linked.resubmitted.0.status`, la même chose mais si jamais il y avait plusieurs demandes créées à partir de l'action, prendre la première
- `linked.1.status`, la même chose que `linked.status` mais en prenant la deuxième demande créée
- `linked.var.foo_string`, un champ de la première demande liée.

On pourrait j'espère simplifier, en partant par exemple `form_links_`, avoir `form_links_resubmitted` (identifiant de l'action obligatoire), et derrière l'objet lié, et donc comme documentation on noterait que si tu veux `form_var_string` de la demande liée tu fais `form_links_resubmitted.form_var_links`.

Et je laisserais à plus tard la réflexion pour accommoder la situation avec plusieurs demandes créées depuis la même action. (dans l'immédiat ma position serait de continuer sur la lancée de `simili-queryset` de `LazyFormDefObjectsManager`, et avoir `form_links_resubmitted_set`, et donc `form_links_resubmitted_set.pending` pour toutes les demandes créées par l'action et encore en attente, etc.).

Ok je comprends mieux le souci d'uniformité, mais `form_links_resubmitted.form_var_links` je ne vois pas trop comment le gérer avec les outils actuels ou alors il faut enchaîner les `CompatibilityNamesDict` pour perpétuer la vieille syntaxe sur des fonctionnalités nouvelles, je ne sais pas si c'est ce qu'on veut plutôt que `form_links_resubmitted.var.links`.

Ce que je peux déjà promettre :

- rendre `varname` obligatoire sur cette action
- faire que `form.links` (sur `LazyFormData`) propose un dico vers ces `varname` et rien d'autre
- faire que `form.links.resubmitted.form@` renvoie un `LazyFormData` sur la première demande créée

Ça supportera la syntaxe `form_links_resubmitted_form` et `form_links_resubmitted_form_var_zob` mais pas `form_links_resubmitted.form_var_zob`.

Les tests ne passaient pas sans ça, mais le code a pu changer depuis.

Normalement on ne devrait passer dans `deep_bytes2str` que les objets pickle anciens, et comme tout ça sera fait avant que cette action ne soit employée, jamais rencontrer ici de tuple ou `namedtuple`.

Sauf que comme remonté `deep_byte2str` est appelé systématiquement et pas seulement au moment d'une migration, donc il doit marcher même sur les nouveaux objets ce qui n'est pas le cas ici (un `namedtuple` est un objet dont la classe est dans `w.c.s.` mais n'a pas de *dict* (d'ailleurs je vais juste faire `hasattr(obj, '__dict__')`).

#56 - 28 janvier 2020 17:34 - Benjamin Dauvergne

Frédéric Péters a écrit :

Faudrait idéalement voir pourquoi ça arrive là-dedans, parce que normalement ça ne devrait pas. (mais sinon, oui passons et je chercherai moi-même plus tard).

Ça vient de là :

```
File "/var/lib/jenkins/workspace/86-Initialisation-d-un-brouillon/wcs/workflows.py", line 477, in __setstate
```

```
475     def __setstate__(self, dict):
476         self.__dict__.update(dict)
> 477         pickle_2to3_conversion(self)
478         for s in self.possible_status + (self.global_actions or []):
479             s.parent = self
```

#57 - 28 janvier 2020 17:51 - Frédéric Péters

Ça supportera la syntaxe `form_links_resubmitted_form` et `form_links_resubmitted_form_var_zob` mais pas `form_links_resubmitted.form_var_zob`.

oui, j'ai utilisé le point pour bien marquer la frontière entre les deux mais c'était accessoire.

Ça vient de là :

ok ça doit s'éviter via

```
def __setstate__(self, dict):
    self.__dict__.update(dict)
- pickle_2to3_conversion(self)
+ if six.PY3 and any(isinstance(k, bytes) for k in o.__dict__):
+     pickle_2to3_conversion(self)
    for s in self.possible_status + (self.global_actions or []):
```

je regarderai ça.

#58 - 28 janvier 2020 18:00 - Benjamin Dauvergne

Un souci que mon implémentation actuelle corrigeait : form_url ne fonctionne pas pour un draft, il y un slash en trop, je vais devoir corriger dans wcs/formdata.py pour que ça passe.

#59 - 28 janvier 2020 18:07 - Frédéric Péters

Un souci que mon implémentation actuelle corrigeait : form_url ne fonctionne pas pour un draft, il y un slash en trop, je vais devoir corriger dans wcs/formdata.py pour que ça passe.

C'est visiblement un autre ticket.

#60 - 28 janvier 2020 21:16 - Benjamin Dauvergne

Frédéric Péters a écrit :

Un souci que mon implémentation actuelle corrigeait : form_url ne fonctionne pas pour un draft, il y un slash en trop, je vais devoir corriger dans wcs/formdata.py pour que ça passe.

C'est visiblement un autre ticket.

Ouaip, j'ai ouvert [#39356](#) mais j'espère le passer avant pour pouvoir garder mes tests comme ils sont.

#61 - 29 janvier 2020 09:53 - Benjamin Dauvergne

Branches à jour, ça inclut la modification proposée pour [#39356](#) (je rebaserai sur la bonne quand ça arrivera) ; aussi la correction à Workflow.__setstate__ au lieu de deep_bytes2str. Toutes les autres modifications demandées sont intégrées (form_links/get_*_fiels inutiles retirés).

Via la modification temporaire de [#39356](#) on peut utiliser simplement form_links.<varname>.form_(backoffice_)url@ et on arrive bien soit sur le draft FO ou le draft BO.

Maintenant il faudrait une relecture complète.

#62 - 30 janvier 2020 10:18 - Serghei Mihai

Relu la branche consolidée. Testé en backoffice et it works.

Petite coquille dans le commentaire:

```
# so we revert to quixote behabiour for adding a line
```

behabiour => behaviour

Donc go pour moi, mais je laisse aussi Frédéric donner son avis.

#63 - 30 janvier 2020 12:17 - Frédéric Péters

- Bloqué par Development #39356: faire en sorte que FormData.get_url retourne la bonne URL pour les brouillons ajouté

#64 - 30 janvier 2020 12:34 - Frédéric Péters

J'ai ajouté un ticket nécessaire avant.

Et j'ai beau avoir proposé vite fait cette partie :

```
- pickle_2to3_conversion(self)
+ if six.PY3 and any(isinstance(k, bytes) for k in self.__dict__):
```

+ pickle_2to3_conversion(self)

je n'en suis pas sûr et crains un existant où l'appel systématique était nécessaire. (et comme mon installation locale est migrée c'est compliqué de faire réapparaître des années d'existant).

Ma solution préférée, réconfortante, serait bien sûr que ce patch n'introduise pas d'utilisation de tuple/namedtuple, là où tout w.c.s. vivait bien avec liste et dictionnaire.

~~

(ce conservatisme dans le code étant ma position générale, et qui explique parfois la difficulté de ces relectures, où je dois laisser de côté ce truc où une fixture contient une classe Fixture, ces advanced= sont fait sans parenthèses ou bool() explicites, etc.).

~~

Il reste aussi des bouts dont je n'ai pas encore compris l'existence, comme le `__eq__` tapé sur `WorkflowStatusItem`.

#65 - 30 janvier 2020 14:56 - Benjamin Dauvergne

Frédéric Péters a écrit :

Ma solution préférée, réconfortante, serait bien sûr que ce patch n'introduise pas d'utilisation de tuple/namedtuple, là où tout w.c.s. vivait bien avec liste et dictionnaire.

Ok.

(ce conservatisme dans le code étant ma position générale, et qui explique parfois la difficulté de ces relectures, où je dois laisser de côté ce truc où une fixture contient une classe Fixture, ces advanced= sont fait sans parenthèses ou bool() explicites, etc.).

Ok je réécris tout ça.

~~

Il reste aussi des bouts dont je n'ai pas encore compris l'existence, comme le `__eq__` tapé sur `WorkflowStatusItem`.

C'est pour le test d'import/export pour l'instant les tests vérifient juste que la sérialisation initiale est égale à elle même dans un cycle sérialisation/désérialisation/sérialisation. Je suis plus rassuré si on y rajoute un test d'égalité par rapport à l'objet de départ. Mais je peux virer ça, d'ailleurs en faisant de Mapping un objet ça ne va plus marcher si bien.

#66 - 30 janvier 2020 15:26 - Benjamin Dauvergne

Voilà c'est modifié.

- Mapping est une classe classique
- `bool(self.varname)` pour `advanced=`
- parenthèses autour des expressions complexes `advanced=`
- `WorkflowStatusItem.__eq__` retiré
- plus de class Fixture

#67 - 04 février 2020 16:52 - Frédéric Péters

Possibilité de pousser une branche rebasée au-dessus de [#39356](#) ?

#68 - 04 février 2020 16:58 - Benjamin Dauvergne

Frédéric Péters a écrit :

Possibilité de pousser une branche rebasée au-dessus de [#39356](#) ?

Ok.

#69 - 04 février 2020 17:04 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Frédéric Péters a écrit :

Possibilité de pousser une branche rebasée au-dessus de [#39356](#) ?

Ok.

Voilà c'est rebasé sur master, j'ai lancé uniquement les tests -k create_formdata et corrigé ce qu'il y avait à corriger; il faut attendre l'exécution de jenkins pour voir si autre chose a pu bouger.

#70 - 04 février 2020 17:09 - Frédéric Péters

- Fichier new-form.png ajouté

Je commence (enfin) à utiliser la branche, je clique sur "valider" mais ça me dit erreur, parce que l'identifiant est obligatoire, devrait être sorti des paramètres avancés.

À cette première validation, n'ayant pas encore pu sélectionner les champs, je resterais sur la page d'édition de l'action.

On peut avoir des champs avec des libellés très longs (je pense par exemple aux cases à cocher "J'accepte ceci et cela", ça casse le layout. (cf capture)

J'essaie alors de choisir un autre formulaire mais le champ "Mappings to new form fields", qui est donc vide, m'affiche "champ obligatoire". (je dirais que ça peut rejoindre le point sur la première validation, quand "mappings to new form fields" est vide, valider mais rediriger vers la page d'édition).

Pas très clair à quoi sert le libellé de l'action.

J'ai testé la création d'un brouillon, je l'ai supprimé, j'ai rejoué l'action, je n'ai pas pu via {{form_links_xxx_form_url}} accéder au nouveau brouillon; vaguement je me dis que ça devrait marcher avec :

```
-         if part.varname == varname:
+         if part.varname == varname and part.formdata:
```

#71 - 04 février 2020 17:26 - Benjamin Dauvergne

Frédéric Péters a écrit :

Je commence (enfin) à utiliser la branche, je clique sur "valider" mais ça me dit erreur, parce que l'identifiant est obligatoire, devrait être sorti des paramètres avancés.

Ok j'ai rendu varname et mappings non requis.

On peut avoir des champs avec des libellés très longs (je pense par exemple aux cases à cocher "J'accepte ceci et cela", ça casse le layout. (cf capture)

Est-ce que je tronque ? Il y a un usage commun ?

J'essaie alors de choisir un autre formulaire mais le champ "Mappings to new form fields", qui est donc vide, m'affiche "champ obligatoire". (je dirais que ça peut rejoindre le point sur la première validation, quand "mappings to new form fields" est vide, valider mais rediriger vers la page d'édition).

Je ne vois pas comment faire cela, sans toucher à tout.

```
# wcs/admin/workflow.
if not form.get_submit() == 'submit' or form.has_errors():
    self.html_top('%s - %s' % (_('Workflow'), self.workflow.name))
    r = TemplateIO(html=True)
    r += htmltext('<h2>%s</h2>') % _(self.item.description)
    r += form.render()
    if self.item.support_substitution_variables:
        r += htmltext('<h3>%s</h3>') % _('Variables')
        r += get_publisher().substitutions.get_substitution_html_table()
    return r.getvalue()
else:
    self.item.submit_admin_form(form)
    self.workflow.store()
    return redirect('..')
```

Je peux modifier cela pour faire retourner une valeur à submit_admin_form ?

Pas très clair à quoi sert le libellé de l'action.

Toutes les actions ou presque ont un libellé mais je peux l'enlever.

J'ai testé la création d'un brouillon, je l'ai supprimé, j'ai rejoué l'action, je n'ai pas pu via {{form_links_xxx_form_url}} accéder au nouveau brouillon; vaguement je me dis que ça devrait marcher avec :

[...]

Ok, branche à jour.

#72 - 04 février 2020 17:39 - Frédéric Péters

Je ne vois pas comment faire cela, sans toucher à tout.

Oui ou avec un truc moche sur un champ cache mis en erreur, laissons ça de côté.

Toutes les actions ou presque ont un libellé mais je peux l'enlever.

Je ne mettrais pas de libellé qui ne sert pas. (dans l'action resubmit le libellé correspondait à un bouton affiché).

#73 - 04 février 2020 19:45 - Benjamin Dauvergne

Frédéric Péters a écrit :

Je ne vois pas comment faire cela, sans toucher à tout.

Oui ou avec un truc moche sur un champ cache mis en erreur, laissons ça de côté.

Je trouve le comportement actuel désagréable sur toutes les actions, mais oui ce serait bien qu'on puisse éditer sans revenir à chaque fois à la liste des actions (avec un bouton "Enregistrer et continuer les modifications" comme dans l'admin Django) c'est très frustrant comme comportement.

Je ne mettrais pas de libellé qui ne sert pas. (dans l'action resubmit le libellé correspondait à un bouton affiché).

Le label sert aussi, il me semble, à documenter le workflow (via `get_line_details()`) mais soit, j'ai retiré.

J'ai aussi réordonné les champs pour mettre mappings en second, ça me semble le plus important après le nom du formulaire cible.

#74 - 04 février 2020 21:01 - Frédéric Péters

Je trouve le comportement actuel désagréable sur toutes les actions (...)

<https://dev.entrouvert.org/projects/wcs/issues/new>

#75 - 05 février 2020 08:21 - Frédéric Péters

On peut avoir des champs avec des libellés très longs (je pense par exemple aux cases à cocher "J'accepte ceci et cela", ça casse le layout. (cf capture)

Est-ce que je tronque ? Il y a un usage commun ?

Il y a clairement quelque chose à faire; à regarder alentours il y a la boîte de sélection des critères qui fait `ellipsis(70)` mais ce n'est pas sur un `<select>`, ici je dirais qu'il y a juste à limiter la largeur du `<select>`; il serait peut-être bon que tu consultes Thomas sur ce point; de manière imparfaite,

```
+++ b/wcs/qommon/static/css/dc2/admin.css
...
+.MappingsWidget td:first-child select {
+   max-width: 300px;
+}
```

#76 - 05 février 2020 09:58 - Benjamin Dauvergne

branche à jour.

#77 - 06 février 2020 09:26 - Benjamin Dauvergne

Je n'ai pas regardé ce que ça donne dans la vue inspect, il faudra que je jette un coup d'oeil à ça.

#78 - 06 février 2020 09:43 - Frédéric Péters

Dans la vue inspect il n'y aura rien, si on y veut quelque chose, genre une ou deux formes minimales, ça doit aller dans `get_static_substitution_variables`; a priori profiter de la boucle

```
# Add variables from evolution parts classes
evolution_parts_classes = set(part.__class__ for evolution in self.evolution or [] for part in
                             evolution.parts or [])
for klass in evolution_parts_classes:
    if hasattr(klass, 'get_substitution_variables'):
        d.update(klass.get_substitution_variables(self))
```

et ajouter un `get_substitution_variables()`.

Le travail d'adaptation de la vue inspect au mode "lazy" sort du cadre du ticket.

~~

Par contre sur le rendu, je pense toujours opportun de demander à Thomas J.

#80 - 06 février 2020 11:13 - Thomas Jund (congé, retour le 29/04)

MappingsWidget utilise un table layout (WidgetListAsTable), il existe des alternatives ou je dois proposer une solution avec le table layout ?

#81 - 06 février 2020 13:45 - Thomas Jund (congé, retour le 29/04)

- Fichier `0001-fixe-MappingsWidget-layout.patch` ajouté

Voici une proposition pour améliorer le layout du MappingsWidget:

1. éviter que le select de l'expression ne passe à la ligne avec un `width + calc()`
2. éviter que le tableau puisse s'agrandir (`width 100%`)
3. fixer une largeur de 40% par défaut au select
4. aligner les labels de colonne à gauche et virer le bold (parce que je préfère)

#82 - 06 février 2020 14:08 - Serghei Mihai

- Fichier `MappingsWidget.png` ajouté

Avec ta proposition le "select" prend la moitié de l'écran même si le libellé du champ est court.

#83 - 06 février 2020 14:40 - Frédéric Péters

Avec ta proposition le "select" prend la moitié de l'écran même si le libellé du champ est court.

Ça n'est pas un problème. (même si perso je serais plutôt pour 30%).

#84 - 06 février 2020 16:51 - Benjamin Dauvergne

- Fichier `Firefox_Screenshot_2020-02-06T15-51-00.908Z.png` ajouté

J'ai ajouté une implémentation de `get_substitution_variables()` pour la vue inspect (copie d'écran attachée) et j'ai intégré le patch de Thomas, branche à jour.

#85 - 07 février 2020 08:27 - Frédéric Péters

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

Ok après un dernier changement, le nom de l'action, New form → New Form Creation. (et après avoir squashed les commits sans numéros dans les commits appropriés)

Et on verra de toute façon des ajustements derrière.

#86 - 07 février 2020 11:46 - Benjamin Dauvergne

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

```
commit 507eaac30b7d7584c5d737d43c2c188ca5e40e0d
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Sat May 18 14:17:33 2019 +0200
```

add new action create-formdata (#33186)

commit 3e8f1fff32fca8c1fb9f5c18ff16a34c12949909
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Fri May 17 20:08:42 2019 +0200

formdata: ease iteration of evolutions parts (#33186)

commit 8eeb8f465a4b104990b2a491f5f05198f8204d26
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Fri May 17 17:52:41 2019 +0200

formdef: ease access to widget fields (#33186)

#87 - 07 février 2020 14:14 - Frédéric Péters

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

#88 - 07 février 2020 14:18 - Pierre Cros

Mes remarques après tests :

- l'action peut s'appliquer sur les formulaires désactivés, pas nécessaire je pense
- Le comportement par défaut est de créer un brouillon, il m'a fallu un moment pour comprendre pourquoi la demande était pas créée, je souhaiterais qu'on change ça. Ou qu'on change l'intitulé de l'action.
- Aucune info n'est affichée. Je serais d'avis qu'il faut, comme pour l'action création de document, pouvoir enregistrer un truc dans l'historique (comprenant l'URL de la nouvelle demande) si on le souhaite.

#89 - 07 février 2020 14:42 - Frédéric Péters

[#39637](#), [#39638](#), [#39639](#)

#90 - 07 février 2020 16:50 - Stéphane Laget

Si l'on souhaite créer une demande sur le même formulaire, on devrait pouvoir le faire sans avoir à définir tous les champs un par un comme sur l'ancienne action re-soumission (sauf si un truc m'a échappé)

#91 - 07 février 2020 16:54 - Benjamin Dauvergne

Il faut ouvrir d'autres tickets, celui-ci est terminé.

#92 - 07 février 2020 16:57 - Frédéric Péters

- Lié à Development #39656: création d'une demande, option pour reprendre tous les champs ajouté

#93 - 07 février 2020 16:58 - Frédéric Péters

- Lié à Development #39637: création d'une demande, ne pas proposer les formulaires désactivés dans la liste ajouté

#94 - 07 février 2020 16:58 - Frédéric Péters

- Lié à Development #39638: création d'une demande ou création d'un brouillon ? ajouté

#95 - 07 février 2020 16:58 - Frédéric Péters

- Lié à Development #39639: création d'une demande, option pour afficher quelque chose dans l'historique ajouté

Fichiers

0004-formdata-ease-iteration-of-evolutions-parts-33186.patch	1,79 ko	18 mai 2019	Benjamin Dauvergne
0005-add-new-action-create-formdata-33186.patch	16,1 ko	18 mai 2019	Benjamin Dauvergne
0003-workflows-ease-filtering-common-fields-of-related-fo.patch	1,79 ko	18 mai 2019	Benjamin Dauvergne
0001-formdef-ease-access-to-widget-fields-33186.patch	959 octets	18 mai 2019	Benjamin Dauvergne
0002-workflows-ease-selecting-related-formdefs-33186.patch	1,41 ko	18 mai 2019	Benjamin Dauvergne
0003-add-new-action-create-formdata-33186.patch	17,2 ko	09 novembre 2019	Benjamin Dauvergne
0002-formdata-ease-iteration-of-evolutions-parts-33186.patch	5,7 ko	09 novembre 2019	Benjamin Dauvergne
0001-formdef-ease-access-to-fields-with-varname-33186.patch	1,26 ko	09 novembre 2019	Benjamin Dauvergne
0003-add-new-action-create-formdata-33186.patch	17,2 ko	27 novembre 2019	Benjamin Dauvergne
0005-add-first-test.patch	4,99 ko	27 novembre 2019	Benjamin Dauvergne
0002-formdata-ease-iteration-of-evolutions-parts-33186.patch	5,7 ko	27 novembre 2019	Benjamin Dauvergne

0004-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch	3,89 ko	27 novembre 2019	Benjamin Dauvergne
0001-formdef-ease-access-to-fields-with-varname-33186.patch	1,26 ko	27 novembre 2019	Benjamin Dauvergne
0003-add-new-action-create-formdata-33186.patch	17,2 ko	28 novembre 2019	Benjamin Dauvergne
0005-add-first-test.patch	4,99 ko	28 novembre 2019	Benjamin Dauvergne
0002-formdata-ease-iteration-of-evolutions-parts-33186.patch	5,7 ko	28 novembre 2019	Benjamin Dauvergne
0004-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch	4,38 ko	28 novembre 2019	Benjamin Dauvergne
0001-formdef-ease-access-to-fields-with-varname-33186.patch	1,26 ko	28 novembre 2019	Benjamin Dauvergne
0003-add-new-action-create-formdata-33186.patch	17,2 ko	23 janvier 2020	Benjamin Dauvergne
0007-second-test.patch	5,7 ko	23 janvier 2020	Benjamin Dauvergne
0006-storage-handle-tuple-likes-in-deep_bytes2str.patch	1008 octets	23 janvier 2020	Benjamin Dauvergne
0005-add-first-test.patch	5,04 ko	23 janvier 2020	Benjamin Dauvergne
0002-formdata-ease-iteration-of-evolutions-parts-33186.patch	5,7 ko	23 janvier 2020	Benjamin Dauvergne
0004-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch	4,38 ko	23 janvier 2020	Benjamin Dauvergne
0001-formdef-ease-access-to-fields-with-varname-33186.patch	1,26 ko	23 janvier 2020	Benjamin Dauvergne
0004-add-new-action-create-formdata-33186.patch	17,2 ko	23 janvier 2020	Benjamin Dauvergne
0007-second-test.patch	5,7 ko	23 janvier 2020	Benjamin Dauvergne
0006-add-first-test.patch	5,05 ko	23 janvier 2020	Benjamin Dauvergne
0001-storage-handle-tuple-likes-in-deep_bytes2str.patch	1008 octets	23 janvier 2020	Benjamin Dauvergne
0003-formdata-ease-iteration-of-evolutions-parts-33186.patch	5,7 ko	23 janvier 2020	Benjamin Dauvergne
0005-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch	4,38 ko	23 janvier 2020	Benjamin Dauvergne
0002-formdef-ease-access-to-fields-with-varname-33186.patch	1,26 ko	23 janvier 2020	Benjamin Dauvergne
0004-add-new-action-create-formdata-33186.patch	17,2 ko	23 janvier 2020	Benjamin Dauvergne
0007-second-test.patch	5,7 ko	23 janvier 2020	Benjamin Dauvergne
0006-add-first-test.patch	5,05 ko	23 janvier 2020	Benjamin Dauvergne
0001-storage-handle-tuple-likes-in-deep_bytes2str.patch	1,02 ko	23 janvier 2020	Benjamin Dauvergne
0003-formdata-ease-iteration-of-evolutions-parts-33186.patch	5,7 ko	23 janvier 2020	Benjamin Dauvergne
0005-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch	4,38 ko	23 janvier 2020	Benjamin Dauvergne
0002-formdef-ease-access-to-fields-with-varname-33186.patch	1,26 ko	23 janvier 2020	Benjamin Dauvergne
0004-add-new-action-create-formdata-33186.patch	17,6 ko	23 janvier 2020	Benjamin Dauvergne
0007-second-test.patch	5,73 ko	23 janvier 2020	Benjamin Dauvergne
0006-add-first-test.patch	5,08 ko	23 janvier 2020	Benjamin Dauvergne
0001-storage-handle-tuple-likes-in-deep_bytes2str.patch	1,02 ko	23 janvier 2020	Benjamin Dauvergne
0003-formdata-ease-iteration-of-evolutions-parts-33186.patch	5,7 ko	23 janvier 2020	Benjamin Dauvergne
0005-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch	1,36 ko	23 janvier 2020	Benjamin Dauvergne
0002-formdef-ease-access-to-fields-with-varname-33186.patch	1,26 ko	23 janvier 2020	Benjamin Dauvergne
0004-add-new-action-create-formdata-33186.patch	17,6 ko	23 janvier 2020	Benjamin Dauvergne
0007-second-test.patch	5,73 ko	23 janvier 2020	Benjamin Dauvergne
0006-add-first-test.patch	5,08 ko	23 janvier 2020	Benjamin Dauvergne
0001-storage-handle-tuple-likes-in-deep_bytes2str.patch	1,02 ko	23 janvier 2020	Benjamin Dauvergne
0009-create_formdata-fix-XML-export-import-of-mappings-wi.patch	1,86 ko	23 janvier 2020	Benjamin Dauvergne
0003-formdata-ease-iteration-of-evolutions-parts-33186.patch	5,7 ko	23 janvier 2020	Benjamin Dauvergne
0005-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch	1,36 ko	23 janvier 2020	Benjamin Dauvergne
0010-tests-add-create_formdata-test-on-XML-export-import-.patch	3,56 ko	23 janvier 2020	Benjamin Dauvergne
0008-workflows-add-WorkflowStatusItem.__eq__-operator-33186.patch	1017 octets	23 janvier 2020	Benjamin Dauvergne
0002-formdef-ease-access-to-fields-with-varname-33186.patch	1,26 ko	23 janvier 2020	Benjamin Dauvergne
0004-add-new-action-create-formdata-33186.patch	17,6 ko	23 janvier 2020	Benjamin Dauvergne
0007-second-test.patch	5,73 ko	23 janvier 2020	Benjamin Dauvergne
0006-add-first-test.patch	5,08 ko	23 janvier 2020	Benjamin Dauvergne
0001-storage-handle-tuple-likes-in-deep_bytes2str.patch	1,02 ko	23 janvier 2020	Benjamin Dauvergne
0009-create_formdata-fix-XML-export-import-of-mappings-wi.patch	1,86 ko	23 janvier 2020	Benjamin Dauvergne
0003-formdata-ease-iteration-of-evolutions-parts-33186.patch	5,7 ko	23 janvier 2020	Benjamin Dauvergne
0005-formdata-make-LinkedFormdataSubstitutionProxy-work-i.patch	1,36 ko	23 janvier 2020	Benjamin Dauvergne
0011-tests-add-create_formdata-admin-page-tests-33186.patch	4,6 ko	23 janvier 2020	Benjamin Dauvergne

0010-tests-add-create_formdata-test-on-XML-export-import-.patch	3,56 ko	23 janvier 2020	Benjamin Dauvergne
0008-workflows-add-WorkflowStatusItem.__eq__-operator-331.patch	1017 octets	23 janvier 2020	Benjamin Dauvergne
0002-formdef-ease-access-to-fields-with-varname-33186.patch	1,26 ko	23 janvier 2020	Benjamin Dauvergne
new-form.png	22,9 ko	04 février 2020	Frédéric Péters
0001-fixe-MappingsWidget-layout.patch	1,11 ko	06 février 2020	Thomas Jund (congé, retour le 29/04)
MappingsWidget.png	190 ko	06 février 2020	Serghei Mihai
Firefox_Screenshot_2020-02-06T15-51-00.908Z.png	12,7 ko	06 février 2020	Benjamin Dauvergne