

## w.c.s. - Development #8273

### données structurées supplémentaires, "champs backoffice"

16 septembre 2015 13:25 - Thomas Noël

<b>Statut:</b>	Fermé	<b>Début:</b>	16 septembre 2015
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>		<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>	v1.47	<b>Planning:</b>	
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
Objectif : permettre d'avoir des informations enregistrées par les agents qui gèrent une demande. Equivalent au "cadre réservé à l'administration".			
Utiles pour :			
<ul style="list-style-type: none"><li>compléter une demande</li><li>«redresser» une information</li><li>préparer des éléments pour une réponse</li></ul>			
Ensemble de champs à définir dans le formdef, sur le même mode/ergonomie que les options de workflows.			
Les champs sont affichés dans la sidebar d'une demande. Un bouton "modifier" est dispo si l'agent est destinataire (ou possède un autre rôle, défini dans le formdef, sur le même principe que les rôles de saisie backoffice ?)			
<b>Demandes liées:</b>			
Lié à w.c.s. - Development #11443: champs backoffice de présentation (titre/e...		<b>Fermé</b>	<b>20 juin 2016</b>
Lié à w.c.s. - Development #11442: champs de présentation (titre/...) dans le...		<b>Fermé</b>	<b>20 juin 2016</b>
Lié à w.c.s. - Development #11441: affichage des champs backoffice obligatoires		<b>Fermé</b>	<b>20 juin 2016</b>
Lié à w.c.s. - Development #11440: champs backoffice non textuels		<b>Fermé</b>	<b>20 juin 2016</b>

### Révisions associées

#### Révision 687a2456 - 20 juin 2016 15:08 - Frédéric Péters

general: add support for backoffice fields (#8273)

### Historique

#### #1 - 16 septembre 2015 13:26 - Thomas Noël

Question:

- on les place dans le formdef ou dans le workflow ? ça pourrait être pertinent dans le workflow (un même workflow défini les champs de traitement pour "n" formulaires)
- ces champs définissent des form\_backoffice\_var\_\* ?
- quid de l'affichage dans les listing ?
- un pre-remplissage des valeurs devrait être effectué lors de la soumission du formulaire

#### #2 - 17 mai 2016 19:14 - Frédéric Péters

- *Sujet changé de champs backoffice à données structurées supplémentaires, "champs backoffice"*

Sur des discussions la semaine dernière :

- Définition des champs au niveau du workflow
- Stockage dans le formdata.data, avec un id préfixé (c'est vraiment ce qui rend facile le reste)
- Méthode get\_all\_fields() dans formdef, qui fait formdef.fields + formdef.workflow.more\_fields, utilisée à la place de formdef.fields où c'est pertinent.
- Action de workflow avec deux paramètres, 1) sélection du champ, 2) formule donnant ce qui ira dedans (expression python, ezt, etc.)

#### #4 - 06 juin 2016 23:18 - Frédéric Péters

- *Fichier 0001-general-add-support-for-backoffice-fields-8273.patch ajouté*

- *Statut changé de Nouveau à En cours*

- Patch proposed changé de Non à Oui

Ça fait tout, il me semble, sauf les afficher (en barre latérale ?).

#### #5 - 09 juin 2016 16:19 - Frédéric Péters

Plutôt les afficher dans un nouveau panneau, entre "Résumé" et "Historique".

#### #6 - 09 juin 2016 16:19 - Frédéric Péters

Appeler la méthode `get_all_fields` plutôt que `get_fields`, pour qu'elle ait moins l'air d'être un simple retour du `self.fields`.

#### #7 - 11 juin 2016 08:34 - Frédéric Péters

- Fichier `0001-general-add-support-for-backoffice-fields-8273.patch` ajouté

Voilà, avec affichage dans un panneau sous le "Résumé" (appelé "Backoffice Data", en français je pensais à "Informations de traitement"). Et quelques tests en plus.

#### #8 - 17 juin 2016 12:16 - Thomas Noël

Il y a un `get_fields` qui traîne dans `sql.py` (ligne 464).

Utiliser `formdef.workflow.get_backoffice_fields` dans `formdef.get_all_fields` ?

A priori il n'y a pas ici de code pour la gestion des variables de substitution ou de l'api : tu imagines ça dans un patch à venir ?

Dans l'action, peut-être imaginer permettre de poser plusieurs valeurs d'un seul coup, comme pour l'action d'édition du profil ?

#### #9 - 17 juin 2016 12:29 - Frédéric Péters

- Fichier `0001-general-add-support-for-backoffice-fields-8273.patch` ajouté

Thomas Noël a écrit :

Il y a un `get_fields` qui traîne dans `sql.py` (ligne 464).

Yep, corrigé en local et oublié d'attacher le nouveau patch.

Utiliser `formdef.workflow.get_backoffice_fields` dans `formdef.get_all_fields` ?

Yep.

A priori il n'y a pas ici de code pour la gestion des variables de substitution ou de l'api : tu imagines ça dans un patch à venir ?

J'avais oublié un `s/fields/get_all_fields/`. Tu penses à quoi pour l'API ?

Dans l'action, peut-être imaginer permettre de poser plusieurs valeurs d'un seul coup, comme pour l'action d'édition du profil ?

ok.

Patch à jour, sans ce dernier point.

#### #10 - 19 juin 2016 11:27 - Frédéric Péters

- Fichier `0001-general-add-support-for-backoffice-fields-8273.patch` ajouté

Voilà avec l'action modifiée pour pouvoir concerner plusieurs champs.

#### #11 - 20 juin 2016 10:50 - Thomas Noël

Frédéric Péters a écrit :

A priori il n'y a pas ici de code pour la gestion des variables de substitution ou de l'api : tu imagines ça dans un patch à venir ?

J'avais oublié un `s/fields/get_all_fields/`. Tu penses à quoi pour l'API ?

Je me demandais si on séparait les variables du formulaire (form\_var\_\*) et celles liées à son traitement (form\_bvar\_\* ?), dans les variables de substitution et/ou dans l'API. Et donc, ma "réflexion":

- pour les variables, laisser des form\_var\_\* est sans doute le plus simple à expliquer et utiliser. Il faudra vérifier qu'en cas d'homonymie du varname, la valeur backoffice écrase la valeur du formulaire (ça pourrait être très pratique pour permettre à un agent d'amender une demande) ;
- pour l'API en revanche, on pourrait avoir un ['workflow']['fields'] séparé.

**#12 - 20 juin 2016 11:07 - Frédéric Péters**

- Fichier 0001-general-add-support-for-backoffice-fields-8273.patch ajouté

pour les variables, laisser des form\_var\_\* est sans doute le plus simple à expliquer et utiliser. Il faudra vérifier qu'en cas d'homonymie du varname, la valeur backoffice écrase la valeur du formulaire (ça pourrait être très pratique pour permettre à un agent d'amender une demande) ;

J'interdirais, par convention, les varnames identiques. (je vois déjà arriver le moment où les champs ne sont pas de même type).

pour l'API en revanche, on pourrait avoir un ['workflow']['fields'] séparé.

Fait dans le patch attaché.

**#13 - 20 juin 2016 14:02 - Thomas Noël**

Frédéric Péters a écrit :

pour les variables, laisser des form\_var\_\* est sans doute le plus simple à expliquer et utiliser. Il faudra vérifier qu'en cas d'homonymie du varname, la valeur backoffice écrase la valeur du formulaire (ça pourrait être très pratique pour permettre à un agent d'amender une demande) ;

J'interdirais, par convention, les varnames identiques. (je vois déjà arriver le moment où les champs ne sont pas de même type).

Ca me va très bien.

pour l'API en revanche, on pourrait avoir un ['workflow']['fields'] séparé.

Fait dans le patch attaché.

Parfait.

Je teste tout cela "à la main".

**#14 - 20 juin 2016 14:12 - Frédéric Péters**

- Fichier 0001-general-add-support-for-backoffice-fields-8273.patch ajouté

**#15 - 20 juin 2016 14:29 - Frédéric Péters**

- Fichier 0001-general-add-support-for-backoffice-fields-8273.patch ajouté

**#16 - 20 juin 2016 15:09 - Frédéric Péters**

- Fichier 0001-general-add-support-for-backoffice-fields-8273.patch ajouté

**#17 - 20 juin 2016 15:38 - Thomas Noël**

Ack.

Notes pour plus tard, tickets à faire au cas où:

- tester les affectations pour les champs de type date, fichier, etc.
- seuls les champs complétés sont affichés en backoffice (page de traitement d'un formdata) : il serait préférable d'afficher tous les champs marqués obligatoires, même s'ils sont vides
- toujours sur la page de traitement d'un formdata, afficher les champs de type titre/sous-titre/commentaires
- en revanche, ne pas lister les champs titre/sous-titre/commentaires dans l'action "Set Backoffice Fields"
- idem, ne pas lister les champs titre/sous-titre/commentaires dans la case "Backoffice Fields (changer)" de définition d'un workflow
- à noter, crash tardif dans sql.py si la formule est « =4+4 » alors que le champs est de type texte

**#18 - 20 juin 2016 15:42 - Frédéric Péters**

- Statut changé de *En cours* à *Résolu* (à déployer)

```
commit 687a2456cea7e5eac42d8fa3ddda4d8a3eadb709
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Mon Jun 6 18:55:21 2016 +0200
```

```
general: add support for backoffice fields (#8273)
```

**#19 - 20 juin 2016 15:42 - Frédéric Péters**

- Lié à *Development* #11443: champs backoffice de présentation (titre/etc.) dans la sélection de l'action "set backoffice fields" ajouté

**#20 - 20 juin 2016 15:43 - Frédéric Péters**

- Lié à *Development* #11442: champs de présentation (titre/...) dans les champs backoffice ajouté

**#21 - 20 juin 2016 15:43 - Frédéric Péters**

- Lié à *Development* #11441: affichage des champs backoffice obligatoires ajouté

**#22 - 20 juin 2016 15:43 - Frédéric Péters**

- Lié à *Development* #11440: champs backoffice non textuels ajouté

**#24 - 28 juin 2016 20:01 - Frédéric Péters**

- Version cible mis à v1.47

**#25 - 04 décembre 2016 13:11 - Frédéric Péters**

- Statut changé de *Résolu* (à déployer) à *Fermé*

**Fichiers**

0001-general-add-support-for-backoffice-fields-8273.patch	19,2 ko	06 juin 2016	Frédéric Péters
0001-general-add-support-for-backoffice-fields-8273.patch	30,4 ko	11 juin 2016	Frédéric Péters
0001-general-add-support-for-backoffice-fields-8273.patch	32,3 ko	17 juin 2016	Frédéric Péters
0001-general-add-support-for-backoffice-fields-8273.patch	36,2 ko	19 juin 2016	Frédéric Péters
0001-general-add-support-for-backoffice-fields-8273.patch	38,5 ko	20 juin 2016	Frédéric Péters
0001-general-add-support-for-backoffice-fields-8273.patch	38,5 ko	20 juin 2016	Frédéric Péters
0001-general-add-support-for-backoffice-fields-8273.patch	39,2 ko	20 juin 2016	Frédéric Péters
0001-general-add-support-for-backoffice-fields-8273.patch	39,2 ko	20 juin 2016	Frédéric Péters