

## w.c.s. - Development #325

### Simplification de l'interface de définition des conditions dans le paramétrage d'un champ "nouvelle page"

14 mars 2011 16:26 - Pierre Cros

<b>Statut:</b> Fermé	<b>Début:</b> 14 mars 2011
<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>	<b>Planning:</b>
<b>Patch proposed:</b> Non	

**Description**

Idéalement il faudrait deux liste déroulantes pour définir une condition : la première permettrait de choisir la question sur laquelle portent la question, et la deuxième la réponse permettant de satisfaire la condition.

Comme j'imagine que la deuxième liste est plus compliquée à mettre en place (puisque dépendant de la sélection dans la première), elle peut-être remplacé par une simple zone de texte qui serait remplie par l'utilisateur. L'important c'est vraiment la première liste.

#### Historique

##### #1 - 17 avril 2011 08:36 - Frédéric Péters

- *Projet changé de Poll-O à w.c.s.*

##### #2 - 19 décembre 2011 00:08 - Thomas Noël

Sauf erreur de ma part, les conditions sont des expressions Python, et peuvent donc être très génériques, surtout depuis qu'on peut y utiliser les variables de substitution (genre "form\_var\_champnom.upper().startswith('T')" pour savoir si un champ commence par un 'T'). Très difficile à rendre, sauf à faire un peu trop cliquodrome.

Je serais d'avis d'avoir une aide en ligne avec quelques exemples classiques pertinents.

##### #3 - 19 décembre 2011 10:50 - Benjamin Dauvergne

[redmine@entrouvert.com](mailto:redmine@entrouvert.com) écrivait:

La demande [#325](#) a été mise à jour par Thomas Noël.

Sauf erreur de ma part, les conditions sont des expressions Python, et peuvent donc être très génériques, surtout depuis qu'on peut y utiliser les variables de substitution (genre "form\_var\_champnom.upper().startswith('T')" pour savoir si un champ commence par un 'T'). Très difficile à rendre, sauf à faire un peu trop cliquodrome.

Je serais d'avis d'avoir une aide en ligne avec quelques exemples classiques pertinents.

Au pire on peut faire une interface de ce genre, mais c'est peut-être du temps passé pour rien si l'aide est suffisante:

```
[ ] Condition simple
  Champ: [ ^ Raison de la demande ]
  Opération: [ ^ contient | égale | terminé par | ]
  Valeur: [ _____ ]

[X] Condition complexe (Expression Python)
  [ _____ ]
```

##### #4 - 01 octobre 2012 16:53 - Victor Claudet

- *Version cible mis à Au-quotidien 2014.5*

##### #5 - 08 avril 2014 11:07 - Thomas Noël

- *Version cible Au-quotidien 2014.5 supprimé*

## #6 - 21 septembre 2016 10:12 - Pierre Cros

- Patch proposed mis à Non

Fred me dit d'écrire dans ce ticket dynamique mes idées de simplification de l'interface (qui vont au-delà) de l'écriture des conditions... Je souhaite masquer la complexité de Python au moins dans l'écriture des variables et je comprends que ça n'est pas facile.

[09:59:42] <pcros> Quand tu dis il n'y a pas moyen, ça veut dire trop coûteux ?

[09:59:51] <fpeters> ça veut dire pas possible.

[10:00:10] <fpeters> on ne peut pas remplacer python par un langage perso.

[10:00:16] <pcros> Parce qu'avoir une couche d'abstraction entre ce qui est saisi et ce qui est communiqué au système, je vois pas ce qu'il y a d'impossible

[10:00:27] <fpeters> on saisit du python.

[10:00:31] <fpeters> on peut remplacer ça par autre chose

[10:00:34] <pcros> Et y a pas mal d'endroits où ce serait utile, pas simplement pour les variables

[10:00:43] <fpeters> mais on va perdre trop de choses.

[10:00:48] <pcros> oui je sais qu'on saisit du Python et c'est ce que je voudrais changer

[10:01:15] <fpeters> donc oui, trop coûteux d'inventer et développer un mini langage.

[10:01:48] <pcros> le mini-langage c'est 10 instructions, hein

[10:01:56] <fpeters> puis la 11ème.

[10:01:59] <fpeters> puis la 12ème.

[10:02:35] <fpeters> on peut définir un sous-ensemble de même pas un langage juste une forme d'expression.

[10:03:15] <pcros> L'autre possibilité c'est de tout rendre graphique/contrôlé

[10:03:17] <fpeters> mais après faut expliquer que soudainement, sur une frontière bizarre, ça devient tout différent et faut faire du python.

[10:04:01] <fpeters> que ça soit du texte ou du graphique, c'est pareil, il y a à définir où se trouve cette frontière.

[10:04:21] <fpeters> on peut travailler à faire l'inventaire exhaustif des expressions aujourd'hui utilisées.

[10:04:46] <pcros> Reprenant l'exemple des variables, je ne peux plus insérer de variables dans un template ou ailleurs qu'en les choisissant dans une liste déroulante

[10:05:02] <fpeters> le problème c'est pas l'ezt.

[10:05:32] <pcros> ok bon tant pis, j'aurai essayé

## #7 - 21 septembre 2016 10:17 - Brice Mallet

[10:04:21] <fpeters> on peut travailler à faire l'inventaire exhaustif des expressions aujourd'hui utilisées.

serait déjà une bonne chose

## #8 - 21 septembre 2016 10:52 - Benjamin Dauvergne

Mon avis c'est qu'on devrait déjà travailler à signaler les erreurs prévisibles au niveau ezt et expressions:

- on devrait pouvoir depuis un workflow obtenir la liste de toutes les variables communes (même nom, même type) à tous les formulaires liés, si pas de formulaire lié uniquement les variables "de base" (celle qu'on a si un formulaire ne contient aucun champ) ; à voir si ça bloque un usage actuel (même nom de variable pour un champ texte dans un formulaire backoffice qui est un champ item dans un formulaire frontoffice);
  - cela implique qu'il faut toujours avoir un formulaire avant de démarrer le développement d'un workflow complexe, sinon on a aucune variable disponible dans le workflow (mais pour des workflow de base/génériques ce n'est pas un souci)
- on devrait pouvoir au niveau d'un workflow ou d'un formulaire lancer un check syntaxique+typage qui vérifie que toutes les expressions et templates ne référencent que des variables définies (il y a une limite bien sûr, on ne sait pas forcément la structure de ce qu'il y a dans une variable et donc on doit s'arrêter au premier déréférencement, i.e. `form_var_XXX_raw` on peut vérifier que c'est possible, `form_var_XXX_raw.zzz` on ne saura qu'au runtime si cela existe)
- on affiche immédiatement au niveau de la fabrication de formulaire ou de la fabrication de workflow les erreurs trouvées avec des liens vers les pages où la correction doit être faite, une erreur ne doit pas rester cachée au fond d'un formulaire ou d'un workflow
- lors de la définition d'un template ou d'une expression on refuse immédiatement toute expression référençant une variable inconnue ou non commune à tous les formulaires ( et on signale si elle est inconnue ou absente de seulement un ou plusieurs formulaires liés)
- on fournit une aide à la saisie pour les champs expression et template, i.e. on propose une liste des variables complètes et à jour (via le mécanisme de la première puce)
- on doit aussi fournir dans l'aide contextuelle des exemples immédiatement utilisables (on explique qu'on peut faire des `or`, `and`, `==`, `[is xx ee]`, etc..)

On peut travailler sur un assistant pour produire des expressions python<sup>1</sup> mais pour moi ça demande la même machinerie que ce que je propose et donc il faut faire ce travail préalable avant d'envisager de faire quoi que ce soit de plus simple.

<sup>1</sup> ce genre de bidule <http://i.stack.imgur.com/nzmgJ.png>

Je dévie mais un côté intéressant à considérer que des formulaires liés à un même workflow ont forcément des variables communes c'est qu'on pourrait côté statistiques exploiter cette logique pour faire des statistiques par workflow (au niveau postgresql, `formdata_XXX` hérite de `formdata_workflow_YYY` qui hérite de `formdata`).