

w.c.s. - Development #6843

le signe == dans les expressions

27 mars 2015 13:05 - Thomas Noël

Statut:	Fermé	Début:	27 mars 2015
Priorité:	Normal	Echéance:	
Assigné à:	Thomas Noël	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Non		
Description			
Quand une expression est foo='bar' au lieu de foo=='bar', on a cette erreur:			
Exception:			
<pre>type = '<type 'exceptions.SyntaxError'>', value = 'invalid syntax (<string>, line 1)'</pre>			
Stack trace (most recent call first):			
<pre>File "/usr/lib/python2.7/dist-packages/wcs/wf/jump.py", line 205, in must_jump 203 condition = self.condition 204 global_variables = get_publisher().get_global_eval_dict() > 205 get_publisher().notify_of_exception(sys.exc_info()) 206 must_jump = False 207</pre>			
(...)			
<pre>condition = 'form_var_intervention="Circulation relative aux travaux"'</pre>			
Dans un premier temps, je propose d'ajouter une note dans la doc pour attirer l'attention sur ce "==".			
Demandes liées:			
Lié à w.c.s. - Development #6844: Gestion des expressions		Fermé	27 mars 2015
Lié à w.c.s. - Development #11042: Avoir un widget validation les expressions...		Fermé	25 mai 2016

Historique

#1 - 27 mars 2015 13:06 - Thomas Noël

- Fichier 0001-help-note-about-in-expressions-6843.patch ajouté
- Statut changé de Nouveau à En cours
- Assigné à mis à Thomas Noël
- Patch proposed changé de Non à Oui

#2 - 27 mars 2015 13:59 - Thomas Noël

Je sais que la question a déjà été évoquée, je ne sais plus où on en était, mais j'imaginerais bien une fonction du genre:

```
def error_in_expression(expression):
    try:
        compile(expression, filename='<string>', mode='eval')
    except SyntaxError, e:
        return _('Erreur de syntaxe au caractère %s') % e.offset
    return None
```

qu'on pourrait appeler quand on affiche un form avec une expression, genre « if error_in_expression(condition): hint="il semble que cette expression comporte une erreur (%s)" % error_in_expression(condition) »

#3 - 27 mars 2015 14:20 - Benjamin Dauvergne

C'est une suite au ticket [#1630](#) il me semble. À ce stade on pourrait même envisager un widget "Expression" qui centraliserait tout ça ? Pour l'instant je crois qu'on a simplement un widget 'Préremplissage' mais c'est un cas particulier du cas général "expression". L'autre source de warning qui serait utile ce serait de pouvoir signaler les variables qui nous semble ne pas exister, en analysant l'arbre de l'expression compilée par rapport aux variables de substitution disponibles. Il y a aussi le fait de pouvoir filtrer les expressions dangereuses genre open('/etc/passwd').read() ou range(1000000000000), la seule bonne façon de faire semble être de faire une whitelist.

#4 - 27 mars 2015 14:21 - Benjamin Dauvergne

Je bouge ma note dans un nouveau ticket j'ai cru celui-ci plus large que ça.

#5 - 27 mars 2015 14:38 - Benjamin Dauvergne

- Lié à *Development #6844: Gestion des expressions ajouté*

#6 - 27 mars 2015 15:06 - Thomas Noël

- Fichier *0001-validate-condition-in-page-field-6843.patch* ajouté

Un exemple de l'idée de la note 2, i.e. juste validation de la syntaxe des expressions lors de leur saisie, et dans ce patch juste pour les conditions de page.

#7 - 27 mars 2015 16:57 - Benjamin Dauvergne

- Fichier *0001-Add-EvalStringWidget-and-use-it-for-page-conditions.patch* ajouté

Je voyais plus un truc comme ça.

#8 - 27 mars 2015 16:59 - Benjamin Dauvergne

Et ensuite évoluer pour que le `EvalStringWidget` ne renvoie pas juste une chaîne mais un objet d'un type `PythonExpression` (qui se pickle bien quand même), ayant une méthode `eval()` et qui implémenterait toutes les siouxeries qu'on pourrait inventer dans le futur.

#9 - 27 mars 2015 19:23 - Thomas Noël

Benjamin Dauvergne a écrit :

Je voyais plus un truc comme ça.

Yep, mieux.

`compile()` peu aussi faire du `TypeError`, on laisse passer ?

(et sinon, je suis tatasse, mais je trouve qu'utiliser `ast.parse` serait plus à propos ici)

#10 - 27 mars 2015 19:55 - Benjamin Dauvergne

Oui on peut remplacer par `ast.parse`.

#11 - 29 mars 2015 11:17 - Frédéric Péters

Je pousserais dès maintenant l'idée vers un `PythonConditionWidget` (qui serait aussi à utiliser pour les conditions dans l'action de workflow "saut") et j'y ajouterais une lecture minimale du résultat de `ast.parse`, pour avertir en cas de `_ast.Assign` que c'est plutôt `==` que la personne veut (et ainsi totalement revenir à l'intitulé de ce ticket).

#12 - 30 mai 2016 15:56 - Thomas Noël

- Lié à *Development #11042: Avoir un widget validation les expressions calculées dans les workflows ajouté*

#13 - 30 mai 2016 15:57 - Thomas Noël

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

fait dans [#11042](#)

#14 - 30 mai 2016 15:57 - Thomas Noël

- Patch proposed changé de *Oui* à *Non*

#15 - 30 mai 2016 18:29 - Frédéric Péters

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

Je viens de mettre un `a = 'b'` dans un champ "condition" de saut et ça n'a pas râlé du tout. D'ailleurs le champ condition est un `StringWidget`.

#16 - 30 mai 2016 22:48 - Thomas Noël

Frédéric Péters a écrit :

Je viens de mettre un a = 'b' dans un champ "condition" de saut et ça n'a pas r al e du tout. D'ailleurs le champ condition est un StringWidget.

J'ai confondu les expression de conditions avec celles pour compute.

#17 - 28 mars 2018 13:34 - Fr ed eric P eters

- Statut chang e de Nouveau   Ferm e

Celui-ci a concern e les expressions; pour l' valuation des conditions il y a maintenant [#22102](#).

Fichiers

0001-help-note-about-in-expressions-6843.patch	1,09 ko	27 mars 2015	Thomas No�l
0001-validate-condition-in-page-field-6843.patch	2,02 ko	27 mars 2015	Thomas No�l
0001-Add-EvalStringWidget-and-use-it-for-page-conditions.patch	2,19 ko	27 mars 2015	Benjamin Dauvergne