

w.c.s. - Development #19112

UI de ComputedExpressionWidget

30 septembre 2017 13:22 - Thomas Noël

Statut:	Fermé	Début:	30 septembre 2017
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		

Description

Actuellement sur un widget "compute" on a le dessin d'un petit cube. Ça n'aide pas encore assez les gens à saisir ce qu'il faut taper, entre ezt ou python...

Ce ticket pour discuter de cette idée d'amélioration :

- on retire le cube, on agrandit la case en largeur
- on affiche « ...[...]... » à la place du petit cube (ou autre idée qui rappellerait ezt)
- on remplace par « Python » quand un "=" est indiqué par l'utilisateur en début de chaîne, et on change de couleur ou on ajoute le logo python jaune/bleu
- on remplace par « ...{{...}}... » quand on accepte du template Django (non je rigole, c'est pour un autre ticket, un jour)

Révisions associées

Révision bbe38a8f - 01 juin 2018 17:53 - Frédéric Péters

general: update UI of expression widget to match condition widget (#19112)

Historique

#1 - 01 octobre 2017 13:48 - Thomas Noël

- Fichier 0001-css-make-expression-s-field-icon-depends-on-expressi.patch ajouté

Juste un test chez moi pour voir ce que ça pourrait donner.

#2 - 02 octobre 2017 23:28 - Frédéric Péters

□□ pour le Python et [...] pour l'ezt ? (pas fan de case trop longue, à voir aussi assurer que la longueur reste fixe sur un changement de type)

#3 - 03 octobre 2017 00:04 - Thomas Noël

- Fichier 2017-10-02_23-48-04.mkv ajouté

- Fichier 0001-css-make-computed-field-icon-depends-on-expression-t.patch ajouté

- Patch proposed changé de Non à Oui

Frédéric Péters a écrit :

□□ pour le Python et [...] pour l'ezt ? (pas fan de case trop longue, à voir aussi assurer que la longueur reste fixe sur un changement de type)

Moi c'est un peu le contraire, je trouve qu'on devrait être assez "explicite" et ne pas utiliser d'icônes qui ne veulent pas dire grand chose aux débutants. Donc je resterais bien sur mon idée de "Python" et "...[...]" avec même, quand c'est vide, un "[.] / Py" pour dire que le champ accepte les deux.

Ok c'est pas top, mais c'est mon idée alors je joue.

Bon c'est vrai que cette case au début peut paraître un peu bizarre... peut-être en la mettant à la fin ?

#4 - 27 mai 2018 15:27 - Frédéric Péters

- Sujet changé de amélioration du pictogramme sur ComputedExpressionWidget à UI de ComputedExpressionWidget

Pour revenir ici, dans la suite de [#24046](#) (UI pour les conditions), je verrais bien quelque chose de similaire ici (sélection explicite du contenu via un bouton sur la droite, avec en grisé dans le champ le type de contenu attendu); pour les conditions le nécessaire a été fait pour conserver de manière

indépendante le format attendu, ça me semble plus compliqué à faire pour ces champs, du coup je composerais avec la contrainte du stockage sous forme de chaîne de caractères.

On pourrait déterminer que le stockage se fait genre "\$format\$ valeur", avec le nécessaire de compatibilité pour comprendre aussi =valeur comme étant \$python\$ valeur, etc.

#5 - 28 mai 2018 09:33 - Thomas Noël

Frédéric Péters a écrit :

Pour revenir ici, dans la suite de [#24046](#) (UI pour les conditions), je verrais bien quelque chose de similaire ici (sélection explicite du contenu via un bouton sur la droite, avec en grisé dans le champ le type de contenu attendu);

Yep.

je composerais avec la contrainte du stockage sous forme de chaîne de caractères.
On pourrait déterminer que le stockage se fait genre "\$format\$ valeur", avec le nécessaire de compatibilité pour comprendre aussi =valeur comme étant \$python\$ valeur, etc.

Ça me semble bien jouable, collision avec l'existant fortement improbable. On aurait donc dans les choix possibles :

- \$django\$... → gabarit Django
- \$ezt\$... → gabarit ezt
- \$python\$... → formule Python (mais qui ne commence plus par un =)
- \$text\$... → texte simple

Et on générerait «tout ce qui n'est pas en \$..\$» via le mécanisme actuel de devinette, pour les calculs mais également pour déterminer l'affichage du widget composite correspondant au type deviné.

#6 - 29 mai 2018 20:10 - Frédéric Péters

- Fichier *0001-general-update-UI-of-expression-widget-to-match-cond.patch* ajouté

- Statut changé de Nouveau à En cours

Après un temps sur cette idée \$type\$ valeur, qui a permis de nettoyer le code, retirer les .startswith('=') à gauche et à droite, je suis revenu en arrière, pour rester sur la même syntaxe de stockage, le seul inconvénient étant de ne pas faire de distinction formelle entre texte simple et gabarit. (dans les avantages il y a l'utilisation pour les champs calculés dans un modèle odt, qui ainsi ne change pas).

Côté code pour montrer que ce n'est rien, je n'ai pas encore retiré la partie qui gérait \$type\$ valeur, il y a ce bout dans form.py :

```
+ # after some time, switching format? (the benefit is to have distinct
+ # modes for text and template).
+ # self.value = '$%s$s' % (value_type, value_content)
```

et ce bout dans workflows.py :

```
+ elif var.startswith('$') and var.count('$') > 1:
+     expression_type, expression_value = var.split('$', 2)[1:]
```

Bref, voici le patch, et je pense que c'est vraiment mieux de faire sans \$type\$ valeur. (et j'en retirerai ces deux morceaux)

#7 - 01 juin 2018 14:58 - Thomas Noël

('Texte' → 'Text' dans le code)

Pourquoi garder 3 choix text/template/python alors qu'en fait c'est seulement template/python qu'on gère finalement ? Je serais presque pour ne garder que ces deux choix. Ou je rate encore qqchose...?

#8 - 01 juin 2018 15:09 - Frédéric Péters

Pourquoi garder 3 choix text/template/python alors qu'en fait c'est seulement template/python qu'on gère finalement ? Je serais presque pour ne garder que ces deux choix. Ou je rate encore qqchose...?

Même si c'est (aujourd'hui) la même chose derrière, ça me semble moins flippant pour l'admin d'entrer du "texte" que de se trouver sur un champ lui demandant un gabarit.

#9 - 01 juin 2018 16:57 - Thomas Noël

- Fichier *Screenshot-2018-6-1 Backoffice de Démarches - Workflow - bo 2 niveau.png* ajouté

Pépin sur le style quand on est sur la gestion des données backoffice, screenshot ci-joint.

#10 - 01 juin 2018 17:24 - Frédéric Péters

- Fichier 0001-general-update-UI-of-expression-widget-to-match-cond.patch ajouté

Corrigée la présence d'une chaîne "Texte" et ce problème de style, j'ai aussi retiré la partie qui aurait géré \$type\$value.

#11 - 01 juin 2018 17:47 - Thomas Noël

Ack

#12 - 01 juin 2018 17:53 - Frédéric Péters

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

```
commit bbe38a8fad115ac4f4afcde98aed8ce1dae7203d
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Mon May 28 14:39:59 2018 +0200
```

```
general: update UI of expression widget to match condition widget (#19112)
```

#13 - 23 décembre 2018 14:45 - Frédéric Péters

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

Fichiers

0001-css-make-expression-s-field-icon-depends-on-expressi.patch	2,23 ko	01 octobre 2017	Thomas Noël
2017-10-02_23-48-04.mkv	291 ko	02 octobre 2017	Thomas Noël
0001-css-make-computed-field-icon-depends-on-expression-t.patch	2,07 ko	02 octobre 2017	Thomas Noël
0001-general-update-UI-of-expression-widget-to-match-cond.patch	19,7 ko	29 mai 2018	Frédéric Péters
Screenshot-2018-6-1 Backoffice de Démarches - Workflow - bo 2 niveau	0,55 ko	01 juin 2018	Thomas Noël
0001-general-update-UI-of-expression-widget-to-match-cond.patch	19,7 ko	01 juin 2018	Frédéric Péters