

w.c.s. - Development #24472

enregistrer les erreurs sur les conditions dans les workflows

12 juin 2018 16:47 - Thomas Noël

Statut:	Fermé	Début:	12 juin 2018
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Suite à #24327 , même chose mais pour les évaluation de conditions sur les actions.			
Demandes liées:			
Précède w.c.s. - Development #24645: enregistrer les erreurs de calcul sur le...		Fermé	13 juin 2018 13 juin 2018

Révisions associées

Révision 25150c21 - 05 juillet 2018 17:29 - Thomas Noël

workflows: only log condition errors as functional errors (#24472)

Historique

#1 - 12 juin 2018 16:59 - Thomas Noël

- Fichier 0001-workflows-only-log-condition-errors-as-functional-er.patch ajouté
- Statut changé de Nouveau à En cours

Voilà le patch, mais je ne mets pas "rustine proposée" encore, parce que le relecteur avisé notera que :

- on ne loggue pas le contexte complet du calcul de la condition. (alors qu'actuellement on a la trace complète, mais "hors sol", liée ni à un formulaire ni à une action)

On pourrait ajouter cette possibilité, mais le lien vers l'inspecteur de la demande qui a planté, et l'action en cause, ça me semble suffisant. (Peut-être pourrait-on plutôt logguer l'id de la dernière demande qui a planté, histoire de regarder l'inspecteur d'une demande qui n'a pas encore trop bougé ?)

- dans les tests, ce moment assert logged_error.occurences_count > 1 # should be 2... == 12 with pickle, 4 with sql

On voit que la modif d'un workflow et le recalcul des droits qui l'accompagne lance plusieurs fois les calculs des conditions. Le code qui fait cette partie me semble cependant déjà pas mal "optimisé", du moins je n'y ai pas encore vu de soucis.

#2 - 19 juin 2018 16:29 - Frédéric Péters

Il me semble ici qu'on risque des summary bien long en y reprenant ainsi le contenu de la condition, ça devrait sans doute aller dans un attribut distinct.

#3 - 19 juin 2018 17:08 - Thomas Noël

Frédéric Péters a écrit :

Il me semble ici qu'on risque des summary bien long en y reprenant ainsi le contenu de la condition, ça devrait sans doute aller dans un attribut distinct.

Je me disais, si la condition ne fait pas partie du summary (ie du slug), alors on ne pas toutes les enregistrer. Bon, dans le cas des conditions sur les actions de workflow, c'est vrai que ça marche, puisque a priori on a une seule condition par action... Si on veut logguer d'autres genre de condition il faudra voir la chose, je pense aux conditions sur les pages voir les champs à venir, sans doute que le tech_id pourra être revu à cette occasion (tiens ça me fait penser que j'ai même pas testé sur les actions globales).

(Sinon, ton avis sur le "assert logged_error.occurences_count > 1" qui montre qu'on calcule peut-être "un peu trop" les conditions ?)

#4 - 19 juin 2018 17:14 - Frédéric Péters

Je me disais, si la condition ne fait pas partie du summary (ie du slug), alors on ne pas toutes les enregistrer. Bon, dans le cas des conditions sur les actions de workflow, c'est vrai que ça marche, puisque a priori on a une seule condition par action... Si on veut logguer d'autres genre de

condition il faudra voir la chose, je pense aux conditions sur les pages voir les champs à venir, sans doute que le tech_id pourra être revu à cette occasion (tiens ça me fait penser que j'ai même pas testé sur les actions globales).

Oui elle était dans la traceback, il faudrait la conserver, mais pas dans le summary, c'est là que je dis qu'il faut un attribut supplémentaire.

(Sinon, ton avis sur le "assert logged_error.occurences_count > 1" qui montre qu'on calcule peut-être "un peu trop" les conditions ?)

Oui, ça sera à regarder.

#5 - 29 juin 2018 15:03 - Thomas Noël

- Fichier 0001-workflows-only-log-condition-errors-as-functional-er.patch ajouté
- Statut changé de En cours à Solution proposée
- Patch proposed changé de Non à Oui

Voici une version qui simplifie le message affiché, qui se résume à "Echec lors du calcul d'une condition (erreur *Exception*)". Quand on clique dessus, quelques détails, la condition en cause, et le message plus complet de l'exception (et bien sûr, comme actuellement, le premier formulaire impacté, le form, le wf, statut, l'action, etc.).

Sur « on calcule peut-être "un peu trop" les conditions » : je n'ai pas réussi à bien voir le soucis encore ; serait plutôt l'objet d'un patch séparé.

#6 - 02 juillet 2018 14:46 - Thomas Noël

- Précède Development #24645: enregistrer les erreurs de calcul sur les champs backoffice (mais ne plus notifier par mail) ajouté

#7 - 02 juillet 2018 15:16 - Frédéric Péters

```
+         r += htmltext(' <li>%s <pre>%s</pre></li>') % (  
+             _('Expression (type %s):') % self.error.expression_type,  
+             self.error.expression)
```

Je trouve que le "Expression (type %s)" donne un rendu bizarre, sans particulièrement chercher à partager avec le code ailleurs (parce qu'il y a des petites différences), j'ajouterais dans l'inspect un expression_titles = {'python': _('Python Expression'), 'django': _('Django Template'), ...}.

Sur la constitution de l'identifiant, on est devenu trop générique je pense, ... failed-to-compute-expression-python-error-nameerror, dommage d'y perdre le nom de la variable. On avait vu un soucis particulier à juste inclure l'exception via %s ?

(de très loin je verrais l'affaire décomposée, on passerait "failed to compute expression" et l'objet exception dans LoggedError il y aurait de quoi sortir quelque chose de précis mais pas trop de l'exception.)

#8 - 03 juillet 2018 10:38 - Thomas Noël

- Fichier 0001-workflows-only-log-condition-errors-as-functional-er.patch ajouté
- Sujet changé de enregistrer les erreurs sur les conditions dans les workflow à enregistrer les erreurs sur les conditions dans les workflows

Bonnes idées, dont voici une implémentation possible.

#9 - 03 juillet 2018 10:56 - Frédéric Péters

```
r += htmltext(' <li>%s <pre>%s</pre></li>') % (  
et
```

et

```
r += htmltext(' <li>%s <pre>%s: %s</pre></li>') % (_('Error message:'),
```

Je taperais <tt> plutôt que <pre>.

```
r += htmltext('<li><a href="logged-errors/%s/">%s</a>') % (error.id, error.summary)  
if error.exception_class or error.exception_message:  
    r += htmltext(_(' error %(class)s (%(message)s)')) % {
```

Plutôt que l'espace en début de chaîne dont je crains la disparition trop facile via la traduction, je l'ajouterais derrière le (ça fera et ça ne portera pas à conséquence).

À part ça, je dirais ok; je me demande juste si à un moment on ne devra pas réfléchir à enregistrer la stack, plutôt pour nous déboguer quand le reste ne suffit pas. (mais laissons ça à plus tard).

#10 - 03 juillet 2018 15:21 - Thomas Noël

- Fichier 0001-workflows-only-log-condition-errors-as-functional-er.patch ajouté

Voilà, avec l'interdiff :

```
diff --git a/wcs/admin/logged_errors.py b/wcs/admin/logged_errors.py
index 542ad5ec..106da872 100644
--- a/wcs/admin/logged_errors.py
+++ b/wcs/admin/logged_errors.py
@@ -78,11 +78,11 @@ class LoggedErrorDirectory(Directory):
     'template': N_('Template'),
     'text': N_('Text'),
     }.get(self.error.expression_type, N_('Unknown'))
-    r += htmltext(' <li>%s <pre>%s</pre></li>' ) % (
+    r += htmltext(' <li>%s %s</li>' ) % (
         _('%s:') % expression_title, self.error.expression)

     if self.error.exception_class or self.error.exception_message:
-        r = htmltext(' <li>%s
%s: %s
</li>' ) % (
r += htmltext(' <li>%s %s: %s</li>' ) % (('Error message:'),
self.error.exception_class,
self.error.exception_message)

@ -175,9 +175,9 @ class LoggedErrorsDirectory(Directory):
'%(count)d error', '%(count)d errors', len(errors)) % {'count': len(errors)}
r = htmltext('<ul>')
for error in errors[:3]:
-    r += htmltext('<li><a href="logged-errors/%s/">%s</a>' ) % (error.id, error.summary)
r = htmltext('<li><a href="logged-errors/%s/">%s</a> ' ) % (error.id, error.summary)
if error.exception_class or error.exception_message:
-    r += htmltext(s (s)') ) % {
r = htmltext(('error %(class)s ((message)s)') ) % {
'class': error.exception_class,
'message': error.exception_message,
}
@ -201,9 +201,9 @ class LoggedErrorsDirectory(Directory):
r += htmltext('<h2>%s</h2>' ) %
r += htmltext('<ul class="biglist">')
for error in self.get_errors(formdef_id=self.formdef_id, workflow_id=self.workflow_id):
-    r += htmltext('<li><strong><a href="%s/">%s</a></strong>' ) % (error.id, error.summary)
r = htmltext('<li><strong><a href="%s/">%s</a></strong> ' ) % (error.id, error.summary)
if error.exception_class or error.exception_message:
-    r += htmltext((' error s ((message)s)') ) % {
r += htmltext(_('error s ((message)s)') ) % {
'class': error.exception_class,
'message': error.exception_message,
}
}
```

À part ça, je dirais ok; je me demande juste si à un moment on ne devra pas réfléchir à enregistrer la stack, plutôt pour nous déboguer quand le reste ne suffit pas. (mais laissons ça à plus tard).

Je me disais de mon côté, enregistrer les 10 derniers (formdef, wf, formdata, status, status_item, datetime) en cause, histoire de ne pas conserver que le premier formdata en cause (qui risque d'avoir bougé sinon disparu après quelques jours) → [#25086](#). Bien d'autres tickets à faire pour améliorer tout ça, oui.

#11 - 05 juillet 2018 16:27 - Frédéric Péters

Ok; tu crées un ticket sur l'idée de conserver une liste des récentes occurrences ?

#12 - 05 juillet 2018 17:30 - Thomas Noël

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

```
commit 25150c2136611da6227e94ea894bf7eaf598efb8
Author: Thomas NOEL <tnoel@entrouvert.com>
Date: Tue Jun 12 16:49:12 2018 +0200
```

workflows: only log condition errors as functional errors (#24472)

#13 - 23 décembre 2018 14:39 - Frédéric Péters

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

Fichiers

0001-workflows-only-log-condition-errors-as-functional-er.patch	7,94 ko	12 juin 2018	Thomas Noël
0001-workflows-only-log-condition-errors-as-functional-er.patch	12,7 ko	29 juin 2018	Thomas Noël
0001-workflows-only-log-condition-errors-as-functional-er.patch	15,7 ko	03 juillet 2018	Thomas Noël
0001-workflows-only-log-condition-errors-as-functional-er.patch	15,8 ko	03 juillet 2018	Thomas Noël