

w.c.s. - Development #24645

enregistrer les erreurs de calcul sur les champs backoffice (mais ne plus notifier par mail)

19 juin 2018 14:46 - Thomas Noël

Statut:	Fermé	Début:	13 juin 2018
Priorité:	Normal	Echéance:	13 juin 2018
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Comme dans #24327 , offrir la possibilité sur compute et l'utiliser pour remonter les pépins de calcul sur les champs backoffice.			
Demandes liées:			
Suit w.c.s. - Development #24472: enregistrer les erreurs sur les conditions ...		Fermé	12 juin 2018

Révisions associées

Révision 7d2e14a1 - 06 juillet 2018 17:10 - Thomas Noël

wf/backoffice_fields: log compute errors as functional errors only (#24645)

Historique

#1 - 19 juin 2018 15:05 - Thomas Noël

- Fichier 0001-wf-backoffice_fields-only-log-condition-errors-as-fu.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Rien de bien spécial à dire, mais deux notify_of_exception en moins, je suis content.

#2 - 19 juin 2018 15:10 - Thomas Noël

Auto-relecture : peut-être que le summary des LoggedError avec le '%(result)r' n'est pas assez stable, puisqu'il génère un slug via simplify(self.summary) et qu'on cherche quand même à ce que ce slug reste unique sur une erreur donnée.

#3 - 19 juin 2018 15:26 - Frédéric Péters

new_value.__class__.__name__, même si les classes ancien style donneront un résultat moins satisfaisant, je préférerais comme dans le /inspect un type(value)

Auto-relecture : peut-être que le summary des LoggedError avec le '%(result)r' n'est pas assez stable, puisqu'il génère un slug via simplify(self.summary) et qu'on cherche quand même à ce que ce slug reste unique sur une erreur donnée.

Effectivement %(result)r mais également %(value)s, ça va multiplier les enregistrements pour une même origine; comme c'est a priori utile d'avoir ces infos, les mettre dans des attributs supplémentaires de LoggedError ?

#4 - 22 juin 2018 08:51 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

Je remet en "En cours" puisque visiblement ce patch n'est pas encore poussable.

#5 - 29 juin 2018 16:33 - Thomas Noël

- Fichier 0001-wf-backoffice_fields-log-compute-errors-as-functiona.patch ajouté
- Statut changé de En cours à Solution proposée

Une version qui utilise les attributs expression, expression_type et error_message ajoutés dans [#24645](#). On a ainsi un summary plus sommaire.

#6 - 02 juillet 2018 14:27 - Frédéric Péters

ajoutés dans [#24645](#)

Mauvaise référence on est dans [#24645](#).

Dans la pratique, est-ce que la plupart des erreurs ce n'est pas plutôt ici :

```
try:
    new_value = self.compute(field['value'], raises=True,
                             formdata=formdata, status_item=self)
except:
    continue
```

(j'essaie en local de taper une erreur qui se trouverait logguée, et j'échoue : une exception quelconque (genre 1/0) se trouve ignorée par le code précédent et un mauvais type (entier pour champ texte) lève quand même une trace).

#7 - 02 juillet 2018 14:31 - Frédéric Péters

```
TypeError: "record() got an unexpected keyword argument 'expression'"
```

Bon en fait c'est parce que cachée il y a cette exception qui survient; ça doit être lié à "ajoutés dans [#24645](#)", mais quel est donc le vrai ticket d'où tirer un patch ?

#8 - 02 juillet 2018 14:45 - Thomas Noël

Frédéric Péters a écrit :

```
TypeError: "record() got an unexpected keyword argument 'expression'"
```

Bon en fait c'est parce que cachée il y a cette exception qui survient; ça doit être lié à "ajoutés dans [#24645](#)", mais quel est donc le vrai ticket d'où tirer un patch ?

Fatigue, désolé, c'était [#24472](#) qui contient le travail de départ sur LoggedErrors, à relire en premier donc.

#9 - 02 juillet 2018 14:46 - Thomas Noël

- *Suit Development #24472: enregistrer les erreurs sur les conditions dans les workflows ajouté*

#10 - 06 juillet 2018 14:21 - Thomas Noël

- *Statut changé de Solution proposée à En cours*

#11 - 06 juillet 2018 15:25 - Thomas Noël

- *Fichier 0001-wf-backoffice_fields-log-compute-errors-as-functiona.patch ajouté*

- *Statut changé de En cours à Solution proposée*

Voici une version plus simple et plus efficace, qui utilise l'enregistrement de l'exception, gagnée dans [#24472](#).

#12 - 06 juillet 2018 16:17 - Frédéric Péters

- *Statut changé de Solution proposée à Solution validée*

J'aurais juste laissé "Template" en minuscule dans "Failed to compute Template", et "Expression" dans "Failed to compute Python Expression".

À part ça, ack.

#13 - 06 juillet 2018 17:11 - Thomas Noël

- *Statut changé de Solution validée à Résolu (à déployer)*

```
commit 7d2e14a1f2645a01bled76fb52019d4817ba75c2
Author: Thomas NOEL <tnoel@entrouvert.com>
Date: Tue Jun 19 15:04:19 2018 +0200
```

```
wf/backoffice_fields: log compute errors as functional errors only (#24645)
```

Fichiers

0001-wf-backoffice_fields-only-log-condition-errors-as-fu.patch	7,12 ko	19 juin 2018	Thomas Noël
0001-wf-backoffice_fields-log-compute-errors-as-functiona.patch	8,79 ko	29 juin 2018	Thomas Noël
0001-wf-backoffice_fields-log-compute-errors-as-functiona.patch	8,34 ko	06 juillet 2018	Thomas Noël