

w.c.s. - Bug #18431

comportement quand une source de données n'existe pas/plus

04 septembre 2017 20:30 - Frédéric Péters

Statut:	Fermé	Début:	04 septembre 2017
Priorité:	Bas	Echéance:	
Assigné à:	Nicolas Roche	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:	Wishlist	Planning:	Non
Patch proposé:	Oui		
Description			
Pour le moment ça plante la page du formulaire en backoffice, ce qui bloque pas mal les choses.			
Il ne faudrait pas planter mais quelle serait la bonne attitude ? Silencieusement ignorer l'erreur (et faire comme si la source de données était vide) ?			
Alternativement ou en complément, gérer lors de l'affichage des champs, champ par champ, la possibilité d'erreur et dans ce cas remplacer le champ en erreur par un message disant ouille ?			
Demandes liées:			
Lié à Publik - Bug #33465: Crash de la l'accès au formulaire si une source de...		Fermé	28 mai 2019
Dupliqué par w.c.s. - Development #25244: Ne pas afficher une 500 sur un form...		Rejeté	12 juillet 2018

Révisions associées

Révision 0085120f - 18 octobre 2019 11:00 - Nicolas Roche

datasources: avoid raising on lost datasource (#18431)

Historique

#1 - 05 septembre 2017 00:22 - Thomas Noël

Frédéric Péters a écrit :

gérer lors de l'affichage des champs, champ par champ, la possibilité d'erreur et dans ce cas remplacer le champ en erreur par un message disant ouille ?

Je verrais bien cela, sur le champ en erreur (et éventuellemnt en gros warning en haut de la page formulaire). Mais j'ai du mal à imaginer comment ça pourrait se faire (ou alors à travers une exception dédiée "HoustonOnAUnProbleme(message)" qui s'afficherait en rouge ?).

En tout cas, ne pas silencieusement ignorer l'erreur.

(Voir aussi [#15163](#), c'est pas la même chose mais c'est lié, pour éviter de se retrouver dans certaines de ces situations)

#2 - 12 juillet 2018 12:07 - Frédéric Péters

- Dupliqué par Development #25244: Ne pas afficher une 500 sur un formulaire qui requière une source de donnée qui n'existe pas. ajouté

#4 - 28 mai 2019 09:47 - Frédéric Péters

- Lié à Bug #33465: Crash de la l'accès au formulaire si une source de donnée liée à celui-ci est supprimée ajouté

#6 - 15 juillet 2019 16:26 - Pierre Cros

- Version cible mis à Wishlist

#7 - 15 juillet 2019 16:28 - Pierre Cros

- Priorité changé de Normal à Haut

#8 - 15 juillet 2019 16:40 - Pierre Cros

- Priorité changé de Haut à Normal

#9 - 09 septembre 2019 10:57 - Pierre Cros

- *Priorité changé de Normal à Bas*

#10 - 09 octobre 2019 10:52 - Nicolas Roche

- *Assigné à mis à Nicolas Roche*

#11 - 09 octobre 2019 20:41 - Frédéric Péters

Tu ne devrais a priori pas avoir à introduire deux nouvelles méthodes et modifier tout un existant pour appeler celles-ci. Et ça ne devrait pas introduire de jeu de piste de méthode qui appelle des méthodes passées en paramètre.

En gros tu as fait le tour, il faut que `data_sources.get_real`, `data_sources.get_items`, `data_sources.get_object`, `data_sources.get_structured_items` n'échouent jamais sur une source de données manquantes, et pour arriver à ça, en regardant vite fait, ça doit pouvoir se passer en ne touchant pas à ces méthodes même mais en remontant d'un cran, sur `NamedDataSource.get_by_slug`.

#12 - 10 octobre 2019 09:15 - Nicolas Roche

- *Fichier 0001-datasources-avoid-error-500-on-lost-datasource-18431.patch ajouté*

- *Statut changé de Nouveau à Information nécessaire*

- *Patch proposed changé de Non à Oui*

en remontant d'un cran, sur `NamedDataSource.get_by_slug`.

J'avais commencé par là, mais je suis descendu pour pouvoir loguer d'où provient la référence à la source de donnée responsable de l'erreur :

```
"data source 'va_etre_detruite' does not exist" from Site opérateur field on Demande d'accès aux châteaux d'eau (cassé exprés) form
```

#13 - 10 octobre 2019 09:22 - Frédéric Péters

Sans doute pas été assez clair, ce patch ne passera pas.

#14 - 10 octobre 2019 13:36 - Nicolas Roche

Compris, je retente ma chance.

#15 - 10 octobre 2019 13:37 - Nicolas Roche

- *Fichier 0001-datasources-avoid-raising-on-lost-datasource-18431.patch ajouté*

- *Statut changé de Information nécessaire à Solution proposée*

#16 - 10 octobre 2019 15:17 - Frédéric Péters

Toujours pas bien fan, toujours l'idée qu'il est possible d'avoir moins d'impact,

Si tout passe au final bien par `get_by_slug`, plutôt que retourner `None` et gérer le cas particulier à n endroits (ce qui est somme toute pareil que laisser le `KeyError` se lever et le gérer à n endroits), je lui ferais retourner une `datasource` qui fonctionne,

```
...
get_logger().warn("data source '%s' does not exist" % slug)
stub = NamedDataSource()
stub.data_source = {'type': 'formula', 'value': '[]'}
return stub
```

ou une variation de l'idée, genre

```
class StubDataSource(object):
    data_source = {'type': 'formula', 'value': []}

...
get_logger().warn("data source '%s' does not exist" % slug)
return StubDataSource()
```

#17 - 10 octobre 2019 19:44 - Nicolas Roche

- *Lié à Development #36808: erreur 500 quand on veut éditer une source de donnée qui n'existe pas ajouté*

#18 - 11 octobre 2019 13:14 - Nicolas Roche

- Fichier 0001-datasources-avoid-raising-on-lost-datasource-18431.patch ajouté

Désolé, je suis passé complètement à côté.

#19 - 11 octobre 2019 13:27 - Nicolas Roche

- Lié à Development #36808: erreur 500 quand on veut éditer une source de donnée qui n'existe pas supprimé

#20 - 17 octobre 2019 11:26 - Frédéric Péters

```
+ 'This datasource is still used in some formularies.'))
```

(on n'a pas "formularies" dans notre vocabulaire), mais passons, mettons plutôt : "This datasource is still used, it cannot be deleted."

Plus important, get_formdefs_of_all_kinds peut retourner des None dans sa liste (cas d'une action "formulaire" pas configurée), et ça plante du coup derrière ainsi,

Exception:

```
type = '<type 'exceptions.AttributeError'>', value = 'NoneType' object has no attribute 'fields''
```

Stack trace (most recent call first):

```
File "/home/fred/src/eo/wcs/wcs/data_sources.py", line 424, in is_used_in_formdef
 422     def is_used_in_formdef(self, formdef):
 423         from .fields import WidgetField
> 424         for field in formdef.fields or []:
 425             data_source = getattr(field, 'data_source', None)
 426             if not data_source:
```

locals:

```
WidgetField = <class 'wcs.fields.WidgetField'>
formdef = None
self = <NamedDataSource 'xxxxxx' id:8>
```

```
File "/home/fred/src/eo/wcs/wcs/admin/data_sources.py", line 38, in is_used
 36     def is_used(self):
 37         for formdef in get_formdefs_of_all_kinds():
> 38             if self.datasource.is_used_in_formdef(formdef):
 39                 return True
 40             return False
```

#21 - 18 octobre 2019 10:28 - Nicolas Roche

Bien reçu,

j'ai pris en compte la première remarque dans [#15163](#),

et la seconde dans [#37058](#).

#22 - 18 octobre 2019 10:56 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

Ok go.

#23 - 18 octobre 2019 11:00 - Nicolas Roche

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

```
commit 0085120f849dbafc8cd6713e06bea866541daecb
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Fri Oct 11 10:38:40 2019 +0200
```

```
datasources: avoid raising on lost datasource (#18431)
```

#24 - 18 octobre 2019 13:15 - Frédéric Péters

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

Fichiers

0001-datasources-avoid-error-500-on-lost-datasource-18431.patch	9,91 ko	10 octobre 2019	Nicolas Roche
0001-datasources-avoid-raising-on-lost-datasource-18431.patch	6,09 ko	10 octobre 2019	Nicolas Roche
0001-datasources-avoid-raising-on-lost-datasource-18431.patch	3,24 ko	11 octobre 2019	Nicolas Roche