

w.c.s. - Development #51591

crash de l'inspect quand le workflow_data comporte foo et foo_bar

02 mars 2021 18:18 - Thomas Noël

Statut:	Fermé	Début:	02 mars 2021
Priorité:	Normal	Echéance:	
Assigné à:	Paul Marillonnet	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposé:	Oui		

Description

cf #51561, on arrive à une trace de ce genre :

Exception:

```
type = '<class 'KeyError'>', value = ''form_workflow_data_foo_bar''
```

Stack trace (most recent call first):

```
File "/usr/lib/python3/dist-packages/wcs/qommon/substitution.py", line 219, in __getitem__
 217             # TypeError will happen if indexing is used on a string
 218             if i == 1:
> 219                 raise KeyError(key)
 220             else:
 221                 parts = parts[i:]
```

locals:

```
__class__ = <class 'wcs.qommon.substitution.CompatibilityNamesDict'>
current_dict = 'yyy'
i = 1
key = 'form_workflow_data_foo_bar'
part = 'bar'
parts = ['bar']
self = {'form': <wcs.variables.LazyFormData object at 0x7f1af45a9a58>, ... 'form_workflow_data_foo_bar': 'xxx', 'form_workflow_data_foo': 'yyy' ...
```

Révisions associées

Révision 20c6163d - 05 mars 2021 15:31 - Paul Marillonnet

inspect: perform thorough namespace resolution in namesdict (#51591)

Historique

#1 - 02 mars 2021 18:22 - Thomas Noël

Mais je ne parviens pas à reproduire le plantage dans nos tests :

```
--- a/tests/backoffice_pages/test_all.py
+++ b/tests/backoffice_pages/test_all.py
@@ -4397,6 +4397,10 @@ def test_inspect_page(pub, local_user):
     upload.receive([b'hello world'])
     formdata.data['4'] = upload
     formdata.user_id = local_user.id
+   formdata.workflow_data = {
+       'foo': 'xxx',
+       'foo_bar': 'yyy',
+   }
     formdata.store()

    from wcs.admin.settings import UserFieldsFormDef
```

se passe sans problème alors que j'imaginai un crash sur la visite de la page d'inspect un peu en dessous :

```
resp = login(get_app(pub)).get(formdata.get_url(backoffice=True), status=200)
resp = resp.click('Data Inspector')
```

Mon analyse du pb doit donc être fausse...

#4 - 02 mars 2021 23:33 - Thomas Noël

J'ai trouvé le cas qui pose problème :

```
--- a/tests/backoffice_pages/test_all.py
+++ b/tests/backoffice_pages/test_all.py
@@ -4397,6 +4397,12 @@ def test_inspect_page(pub, local_user):
     upload.receive(['hello world'])
     formdata.data['4'] = upload
     formdata.user_id = local_user.id
+   formdata.workflow_data = {
+     'foo': {
+       'bar_coin': 'yy',
+     },
+     'foo_bar': 'xx',
+   }
     formdata.store()
```

Crash du test :

```
[2021-03-02 23:31:17] exception caught
Exception:
  type = '<class 'KeyError'>', value = ''form_workflow_data_foo_bar_coin''
```

Stack trace (most recent call first):

```
File "/home/thomas/src/wcs/wcs/qommon/substitution.py", line 219, in __getitem__
 217         # TypeError will happen if indexing is used on a string
 218         if i == 1:
> 219             raise KeyError(key)
 220         else:
 221             parts = parts[i:]
```

```
locals:
  __class__ = <class 'wcs.qommon.substitution.CompatibilityNamesDict'>
  current_dict = 'xx'
  i = 1
  key = 'form_workflow_data_foo_bar_coin'
  part = 'coin'
  parts = ['coin']
  self = {'form': <wcs.variables.LazyFormData object at 0x7f4dacbd7700>, 'attachments': <wcs.workflows.AttachmentsSubstitutionProxy object at 0x7f4dacbd7400>, 'foo_bar': 'xx', 'foo_bar_coin': 'yy'}
```

#5 - 03 mars 2021 16:24 - Paul Marillonnet

- Fichier 0001-inspect-perform-thorough-namespce-resolution-in-nam.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

#6 - 03 mars 2021 16:35 - Frédéric Péters

(jenkins va au moins péter sur black)

#7 - 03 mars 2021 16:37 - Paul Marillonnet

Sur jabber :

- « – ouch... parts = parts[i:] dans le for i in range(len(parts), 0, -1) ... ça fait peur non ?
- il est dans le if i == 1, donc c'est la dernière itération. je peux ajouter un break si tu veux.
- si c'est possible c'est bien. »

#8 - 03 mars 2021 16:49 - Paul Marillonnet

- Fichier 0001-inspect-perform-thorough-namespce-resolution-in-nam.patch ajouté

Sur jabber encore un bout de relecture :

je remplacerais bien le "if not nest_level:" par "if nest_level > 0:" qui me semblerait plus explicite/clair. et « parts = parts[i:] » par « parts = parts[1:] ».

Edit: et ce nouveau patch conforme à black paramétré selon notre config (py37, pas de normalisation des chaînes, longueur de ligne à 110).

#9 - 03 mars 2021 17:01 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

- Assigné à mis à Paul Marillonnet

(Et en fait, ayant des faux négatifs dans mes tests en local, je n'ai pas vu que ça casse d'autres choses. Patch à revoir donc.)

#10 - 03 mars 2021 17:30 - Paul Marillonnet

Et de façon plus générale on pourrait réfléchir si on veut vraiment aller vers ça, i.e. avec quelque chose comme :

```
+ formdata.workflow_data = {
+   'foo': {
+     'bar_coin': 'toto',
+   },
+   'foo_bar': {
+     'coin': 'tata',
+   },
+ }
```

alors situation ambiguë et on peut avoir des surprises si on n'a plus en tête l'ordre de parcours emprunté par l'algo.

Mais bon ce choix arbitraire survient forcément puisque dans la clé le délimiteur entre espaces de nommage successifs puis variable est un caractère autorisé dans un nom de variable.

Alternativement on pourrait faire en sorte d'empêcher qu'on se retrouve avec des espaces de nommage partageant un préfixe commun ('foo' dans l'exemple ici.)

#11 - 03 mars 2021 17:35 - Thomas Noël

Paul Marillonnet a écrit :

Et de façon plus générale on pourrait réfléchir si on veut vraiment aller vers ça, i.e. avec quelque chose comme :

Oui en cas d'ambiguïté il faudra trouver/planter/choisir-au-pif, mais là je suis chez un client dans une situation sans ambiguïté et avec ce bogue.

Alternativement on pourrait faire en sorte d'empêcher qu'on se retrouve avec des espaces de nommage partageant un préfixe commun ('foo' dans l'exemple ici.)

Dans mon cas c'est la réponse d'un webservice qui renvoie du JSON structuré valable et logique... c'est nous décidons de l'aplatir et créons la difficulté. Le problème n'existait pas à l'origine, il est arrivé avec les optimisations lazy (parfaitement nécessaires).

Bref, va falloir creuser :)

#12 - 03 mars 2021 20:37 - Frédéric Péters

(reste possible d'utiliser des points, si ça a une espèce d'urgence, c'est-à-dire ça va au-delà de l'affichage de page /inspect qui ne marche pas).

#14 - 04 mars 2021 02:48 - Benjamin Dauvergne

C'est un algorithme de recherche avec retour en arrière qui est nécessaire, parce que là on descend linéairement du moins profond au plus profond en prenant la solution la plus courte longue à chaque niveau (si foo matche je descend dans foo, à mince y a pas bar dedans je m'arrête, au lieu de remonter d'un cran pour trouver foo_bar).

J'ai poussé une branche avec ce fonctionnement. J'ai gardé l'ordre actuel du préfixe le plus long au plus court, mais je pense que ce serait plus logique dans l'autre sens ([['form', 'var', 'truc_muche']], plutôt que [['form_var_truc_muche'], ['form_var_truc', 'muche'], ['form_var', 'truc_muche'], ..., ['form', 'var', 'truc_muche']], à vrai dire l'ordre idéal ce serait le plus court, le plus long puis les autres préfixes.

Donc actuellement {'foo_bar': {'coin': 1}} va cacher {'foo': {'bar_coin': 2}}.

#16 - 04 mars 2021 08:37 - Frédéric Péters

Benjamin,

```
- # TypeError will happen if indexing is used on a string
```

je garderais ce commentaire, il est là parce que j'ai trouvé utile rappel que le TypeError qu'on attrape vient de ce moment.

#17 - 04 mars 2021 08:41 - Benjamin Dauvergne

Frédéric Péters a écrit :

je garderais ce commentaire, il est là parce que j'ai trouvé utile rappel que le TypeError qu'on attrape vient de ce moment.

Ok, branche à jour.

#18 - 04 mars 2021 10:01 - Paul Marillonnet

- Statut changé de *En cours* à *Solution validée*

Ok version récursive qui corrige le bogue et rend le code à mon avis bien plus clair. Top.

Peut-être juste documenter quelque part cette précédence de `{'foo_bar': {'coin': 1}}` sur `{'foo': {'bar_coin': 2}}` (résolution 'greedy' du namespace) ?

#19 - 05 mars 2021 15:31 - Benjamin Dauvergne

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

```
commit 20c6163de878f4e1d8b913d896d5f6e1f6dc87bf
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Wed Mar 3 16:21:48 2021 +0100
```

```
inspect: perform thorough namespace resolution in namesdict (#51591)
```

#20 - 08 mars 2021 19:16 - Frédéric Péters

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

Fichiers

0001-inspect-perform-thorough-namespace-resolution-in-nam.patch	2,16 ko	03 mars 2021	Paul Marillonnet
0001-inspect-perform-thorough-namespace-resolution-in-nam.patch	2,2 ko	03 mars 2021	Paul Marillonnet