

w.c.s. - Bug #63816

accès, dans le payload d'un appel ws, à une donnée calculée associée à une valeur de champ fiche et absence de unlazy (?) ("Object of type LazyFieldVar is not JSON serializable")

11 avril 2022 18:33 - Paul Marillonnet

Statut:	Fermé	Début:	11 avril 2022
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Ici en tentant de modifier le champ SIRET de la fiche :
<https://formulaires-moncompte-rec.grandlyon.com/backoffice/data/mandataire/1/>
La modification est bien prise en compte, c'est l'appel ws lié, déclenché sur édition de la fiche qui plante, i.e.
<https://formulaires-moncompte-rec.grandlyon.com/backoffice/workflows/7/status/recorded/items/3/>

C'est le form_var_entreprise_live_var_nom qui semble mal apprécié ici, d'ailleurs côté journalctl -u wcs je vois que le champ correspondant dans la charge utile POST est toujours sous la forme d'un LazyFieldVar :

```
Exception:
  type = '<class 'TypeError'>', value = 'Object of type LazyFieldVar is not JSON serializable'
Stack trace (most recent call first):
  File "/usr/lib/python3.7/json/encoder.py", line 179, in default
    177
    178     """
> 179     raise TypeError(f'Object of type {o.__class__.__name__} '
    180                       f'is not JSON serializable')
    181
locals:
  o = <wcs.variables.LazyFieldVar object at 0x7f7a180ccd30>
  self = <wcs.qommon.misc.JSONEncoder object at 0x7f7a0f3a8a90>
File "/usr/lib/python3/dist-packages/wcs/qommon/misc.py", line 635, in default
  633
  634     # Let the base class default method raise the TypeError
> 635     return json.JSONEncoder.default(self, o)
  636
  637
locals:
  o = <wcs.variables.LazyFieldVar object at 0x7f7a180ccd30>
  self = <wcs.qommon.misc.JSONEncoder object at 0x7f7a0f3a8a90>
File "/usr/lib/python3.7/json/encoder.py", line 257, in iterencode
  255         self.key_separator, self.item_separator, self.sort_keys,
  256         self.skipkeys, _one_shot)
> 257     return _iterencode(o, 0)
  258
  259 def _make_iterencode(markers, _default, _encoder, _indent, _floatstr,
locals:
  _encoder = <built-in function encode_basestring_ascii>
  _iterencode = <_json.Encoder object at 0x7f7a0f45c250>
  _one_shot = True
  floatstr = <function JSONEncoder.iterencode.<locals>.floatstr at 0x7f7a0f45dc80>
  markers = {140162218301048: {'email': 'testcutpro03_eo@yopmail.com', 'data': {'entreprise_sir
et': '44317013900036', 'entreprise_nom': <wcs.variables.LazyFieldVar object at 0x7f7a180ccd30>}},
140162218301120: {'entreprise_siret': '44317013900036', 'entreprise_nom': <wcs.variables.LazyField
Var object at 0x7f7a180ccd30>}, 140162366229808: <wcs.variables.LazyFieldVar object at 0x7f7a180cc
d30>}
  o = {'email': 'testcutpro03_eo@yopmail.com', 'data': {'entreprise_siret': '44317013900036', '
entreprise_nom': <wcs.variables.LazyFieldVar object at 0x7f7a180ccd30>}}
  self = <wcs.qommon.misc.JSONEncoder object at 0x7f7a0f3a8a90>
```

Révisions associées

Révision 5c45355d - 15 avril 2022 08:02 - Frédéric Péters

misc: unlazy when serializing json (#63816)

Historique

#1 - 11 avril 2022 18:40 - Frédéric Péters

- Privée changé de Oui à Non

#2 - 11 avril 2022 18:44 - Frédéric Péters

- Assigné à mis à Frédéric Péters

Tu dois pouvoir appliquer |stringformat en attendant.

A priori un unlazy dans le sérialiseur json suffirait. Je tape une branche avec ça, pour déjà voir comment ça se comporte, y ajouter un test plus tard.

#3 - 11 avril 2022 19:03 - Paul Marillonnet

Frédéric Péters a écrit :

Tu dois pouvoir appliquer |stringformat en attendant.

Ah oui bien sûr, et petit détail qui n'en est pas un et que j'avais complètement zappé à l'écriture de ce ticket, c'est une expression python, pas une expression de gabarit django.

J'ai tapé un str(form_var_entreprise_live_var_nom) et ça fonctionne, merci.

#4 - 11 avril 2022 20:09 - Frédéric Péters

- Fichier 0001-misc-unlazy-when-serializing-json-63816.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Ok sans faire de test avec une expression python, sans chercher non plus comment produire la même situation via un gabarit django (je dirais que pas possible), mais le patch basique.

#5 - 12 avril 2022 01:11 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

#6 - 15 avril 2022 08:03 - Frédéric Péters

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

```
commit 5c45355d7e0a6d3a8e228a849d93434dfe1dd1fe
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Mon Apr 11 18:43:09 2022 +0200
```

```
misc: unlazy when serializing json (#63816)
```

#7 - 15 avril 2022 21:16 - Transition automatique

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

#8 - 19 juin 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-misc-unlazy-when-serializing-json-63816.patch

837 octets

11 avril 2022

Frédéric Péters