w.c.s. - Development #41195

gestion de err=1 dans les sources de données JSON

31 mars 2020 14:34 - Thomas Noël

Statut:	Fermé	**Début:**	31 mars 2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Thomas Noël	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Actuellement quand une source de données répond, on analyse pas l'éventuel code err:1 qui peut être envoyé en plus de "data".
Donc :

- si on reçoit une réponse {err:1, data:None} on fait un "Error reading JSON data source" qui n'est pas tout à fait exact
- si on reçoit une {err:1, data:[]} on risque de faire un cache de la donnée reçue (si la source de donnée a un cache configuré)

Ce se passe dans wcs/data_sources.py dans get_structured_items :

```
        try:
            entries = misc.json_loads(misc.urlopen(url).read())
            if type(entries) is not dict:
                raise ValueError('not a json dict')
            if type(entries.get('data')) is not list:
                raise ValueError('not a json dict with a data list attribute')
```

il faudrait selon moi avoir une ValueError déclenchée si err existe et n'est ni 0 ni "0"

Révisions associées

Révision 04b60948 - 03 avril 2020 11:07 - Thomas Noël

data_source: handle err in JSON outputs (#41195)

Historique

#1 - 31 mars 2020 14:47 - Thomas Noël

- Fichier 0001-data_source-handle-err-in-JSON-output-41195.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#2 - 31 mars 2020 14:59 - Frédéric Péters

```
-            if entries.get('err') and entries['err'] not in (0, "0"):
+            if entries.get('err') not in (None, 0, "0"):
```

?

#3 - 31 mars 2020 15:01 - Frédéric Péters

Aussi, tant qu'à être dans les alentours, on a reçu ça :

```
Stack trace (most recent call first):
  File "/usr/lib/python3/dist-packages/wcs/data_sources.py", line 394, in get_structured_value
   392             response = misc.json_loads(resp)
   393             if response['data']:
>  394                 value = response['data'][0]
   395         else:
   396             structured_items = get_structured_items(self.data_source, mode='lazy')
```

```
  locals:
     option_id = '0001'
     resp = b'{"err_class": "passerelle.utils.jsonresponse.APIError", "err_desc": "error status:500 \'Erreur I
```

```
nterne de Servlet\',
content:\'[{\\"logref\\":\\"c0b382fa-0b3f-4776-bae1-237e8f9e2e6a\\",\\"message\\":\\"Erreur non g\\\\xc3\\\\xa
9r\\\\xc3\\\\xa9e par une
application cliente: Failed to convert value of type \\\\\'java.lang.String\\\\\' to required type \\\\\'java.
lang.Long\\\\\'; nested exception
is java.lang.NumberFormatException: For input string: \\\\\\\\\\"None\\\\\\\\\\"\\",\\"links\\":[]}]\'", "data
": {"status_code": 500,
"json_content": [{"message": "Erreur non g\\u00e9r\\u00e9e par une application cliente: Failed to convert valu
e of type \'java.lang.String\' to
required type \'java.lang.Long\'; nested exception is java.lang.NumberFormatException: For input string: \\"No
ne\\"", "links": [], "logref":
"c0b382fa-0b3f-4776-bae1-237e8f9e2e6a"}]}, "err": 1}'
     response = {'err_class': 'passerelle.utils.jsonresponse.APIError', 'data': {'status_code': 500, 'json_con
tent': [{'links': [], 'logref':
'c0b382fa-0b3f-4776-bae1-237e8f9e2e6a', 'message': 'Erreur non gérée par une application cliente: Failed to co
nvert value of type
\'java.lang.String\' to required type \'java.lang.Long\'; nested exception is java.lang.NumberFormatException:
 For input string: "None"'}]},
'err': 1, 'err_desc': 'error status:500 \'Erreur Interne de Servlet\',
content:\'[{"logref":"c0b382fa-0b3f-4776-bae1-237e8f9e2e6a","message":"Erreur non g\\xc3\\xa9r\\xc3\\xa9e par
une application cliente: Failed to
convert value of type \\\'java.lang.String\\\' to required type \\\'java.lang.Long\\\'; nested exception is ja
va.lang.NumberFormatException: For
input string: \\\\"None\\\\"","links":[]}]\''}
     self = <NamedDataSource 'Solis : rues de la commune du bénéficiaire (form_var_beneficiaire_commune)' id:5
>
     value = None
```

#4 - 31 mars 2020 17:11 - Thomas Noël

- Fichier 0001-data_source-handle-err-in-JSON-outputs-41195.patch ajouté

J'ai tenté au départ de factoriser le code mais en fait ça ne rendait pas quelque chose de bien lisible à mon goût, donc c'est un poil dupliqué entre get_structured_items et load_json (qui n'est utilisé que par get_structured_value)

#5 - 01 avril 2020 11:41 - Frédéric Péters

```
-            resp = self.load_json(self.id_parameter, option_id)
-            response = misc.json_loads(resp)
-            if response['data']:
-                value = response['data'][0]
+            value = self.load_json(self.id_parameter, option_id)
```

On passe de value = response['data'][0] à value = la réponse, il y a un truc que je rate.

(...)

```
            resp = misc.urlopen(url).read()
            resp = misc.json_loads(resp)
...
            resp = resp['data'][0]
```

Ça m'irait d'avoir le nom de variable qui change selon son contenu, plutôt que ce recyclage.

Aussi, par rapport à la mise en cache, elle me fait un peu peur, on va avoir dans ce code :

```
request.datasources_cache[unsigned_url] = resp  # {"id": ..., "text": ...}
```

mais dans get_structured_items,

```
                request.datasources_cache[unsigned_url] = items  # [{"id": ..., "text": ...}]
```

(oui il me semble qu'il y avait déjà incohérence avant, avec du premier côté {"data": [{"id":..., "text":...]} qui était mis en cache).

~~

```
+            if type(resp) is not dict:
```

En 2020 on utilise isinstance :)

#6 - 01 avril 2020 11:53 - Thomas Noël

- Statut changé de Solution proposée à En cours

Frédéric Péters a écrit :

> [...]
>
> On passe de value = response['data'][0] à value = la réponse, il y a un truc que je rate.

Ouaip j'ai changé load_json en fait, un peu violemment, elle fait d'ailleurs plus que loader le json, elle l'analyse et extrait le premier item. Je vais la renommer get_item_by_id

> Ça m'irait d'avoir le nom de variable qui change selon son contenu, plutôt que ce recyclage.

Yep.

> Aussi, par rapport à la mise en cache, elle me fait un peu peur, on va avoir dans ce code ... / mais dans get_structured_items ...

Ah, bien vu. Je vais harmoniser cela.

> En 2020 on utilise isinstance :)

Oh hé. Hein. Bon.

#7 - 01 avril 2020 14:39 - Thomas Noël

- Fichier 0001-data_source-handle-err-in-JSON-outputs-41195.patch ajouté

- Statut changé de En cours à Solution proposée

Voici une version plus propre (à mon goût) avec un peu de factorisation, quand même. On cache toujours la liste des items "normés".

#8 - 03 avril 2020 10:14 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

#9 - 03 avril 2020 11:07 - Frédéric Péters

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

```
commit 04b609481ff7569c0f3b2c43c2cc5c0f809d9d8c
Author: Thomas Noël <tnoel@entrouvert.com>
Date:   Tue Mar 31 14:45:41 2020 +0200

    data_source: handle err in JSON outputs (#41195)
```

#10 - 03 avril 2020 12:15 - Frédéric Péters

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

Fichiers

0001-data_source-handle-err-in-JSON-output-41195.patch	2,51 ko	31 mars 2020	Thomas Noël
0001-data_source-handle-err-in-JSON-outputs-41195.patch	7,34 ko	31 mars 2020	Thomas Noël
0001-data_source-handle-err-in-JSON-outputs-41195.patch	10 ko	01 avril 2020	Thomas Noël