

## w.c.s. - Bug #38981

### tentative de sérialisation time.struct\_time

15 janvier 2020 11:52 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	15 janvier 2020
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>		<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
Exception: type = '<type 'exceptions.AssertionError'>', value = 'time.struct_time should not be serialized'			
Stack trace (most recent call first): File "/usr/lib/python2.7/dist-packages/wcs/qommon/misc.py", line 481, in default 479 # default serializer in Python 3 (they are tuples) and should 480 # be converted to datetime objects beforehand. > 481 assert not isinstance(obj, time.struct_time), 'time.struct_time should not be serialized' 482 483 if isinstance(obj, datetime.datetime):  locals: obj = time.struct_time(tm_year=2020, tm_mon=2, tm_mday=2, tm_hour=0, tm_min=0, tm_sec=0, tm_wday=6, tm_yday=33, tm_isdst=-1) self = <wcs.qommon.misc.JSONEncoder object at 0x7f6a520fb7d0>			
La valeur vient d'un  'creche_var_date_raw': time.struct_time(tm_year=2020, tm_mon=2, tm_mday=2, tm_hour=0, tm_min=0, tm_sec=0, tm_wday=6, tm_yday=33, tm_isdst=-1),  creche_var_date pointe un champ "Date" d'un formulaire de workflow.			

#### Révisions associées

##### Révision efa3ebb3 - 15 janvier 2020 16:54 - Frédéric Péters

misc: change json encoder to preprocess date for time.struct\_time (#38981)

#### Historique

##### #1 - 15 janvier 2020 14:49 - Frédéric Péters

Et ailleurs,

```
'form_option_date_ceremonie': time.struct_time(tm_year=2019, tm_mon=9, tm_mday=28, tm_hour=0, tm_min=0, tm_sec=0, tm_wday=5, tm_yday=271, tm_isdst=-1),
```

Je pense que je dois regarder ça et que ça va finir en hotfix :/

##### #2 - 15 janvier 2020 15:43 - Frédéric Péters

- Fichier 0001-misc-change-json-encoder-to-preprocess-date-for-time.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Le premier c'est dans une action d'appel webservice, le second dans un retour d'API; correction au niveau de l'encodeur.

(test uniquement fait pour le cas "API" qui était le plus simple à reproduire)

##### #3 - 15 janvier 2020 16:11 - Thomas Noël

Je ne suis pas fan de l'approximation

```
if dt.hour == 0 and dt.minute == 0 and dt.second == 0:  
    return dt.date()
```

On peut vraiment pas envoyer quand même un datetime à ce moment ? (parce qu'un cron nocturne qui va modifier un truc à 0h0min0s ça va arriver sur telle ou telle demande, et boum le json aura une date au lieu d'un datetime)

(Bon, en fait, je vois pas la cause de nouveau bogue)

#### #4 - 15 janvier 2020 16:19 - Frédéric Péters

On a uniquement des champs date, pas date/heure, donc on aura toujours 00h00m00, cf les deux exemples ici. Pour les receipt\_time, last\_update\_time, etc. j'ai conservé la gestion explicite qui était en place et prend tout :

```
'receipt_time': datetime.datetime(*filled.receipt_time[:6]),  
'last_update_time': datetime.datetime(*filled.last_update_time[:6]),  
...  
'time': datetime.datetime(*self.time[:6]) if self.time else None,  
'last_jump_datetime': self.last_jump_datetime
```

Donc là, je pense l'approximation utilisée assez correcte.

(Bon, en fait, je vois pas la cause de nouveau bogue)

C'est en fait un bug préventif; en Python 3 le time.struct\_time est transformé par la sérialisation json en tuple, on se trouve donc dans l'API avec quelque chose comme "creche\_var\_date\_raw": [2020, 1, 15, 16, 17...]; de manière tout à fait silencieuse.

Et comme c'est vu comme un tuple, c'est pris dans le code d'encodage natif json et on n'a même pas la possibilité de changer le comportement.

Donc prévention du bug silencieux qui serait arrivé, j'ai tapé un assert, pour assurer qu'il n'y avait jamais du time.struct\_time passant à cet endroit.

(alors oui je suis un boulet j'aurais du taper un notify\_of\_exception() et laisser l'affaire tourner, histoire de ne pas crasher)

#### #5 - 15 janvier 2020 16:40 - Thomas Noël

- *Statut changé de Solution proposée à Solution validée*

Frédéric Péters a écrit :

(alors oui je suis un boulet j'aurais du taper un notify\_of\_exception() et laisser l'affaire tourner, histoire de ne pas crasher)

Tout s'éclaire, merci.

Ok pour l'approximation, l'encodeur JSON générique n'est pas en jeu sur les formdata.receipt\_time, evo.time et autres, donc ça va aller.

#### #6 - 15 janvier 2020 16:55 - Frédéric Péters

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

```
commit efa3ebb305ce7b19f63550119cac59c9c5673f0c  
Author: Frédéric Péters <fpeters@entrouvert.com>  
Date: Wed Jan 15 15:42:03 2020 +0100
```

```
misc: change json encoder to preprocess date for time.struct_time (#38981)
```

#### #7 - 15 janvier 2020 18:15 - Frédéric Péters

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

### Fichiers

---

0001-misc-change-json-encoder-to-preprocess-date-for-time.patch	4,22 ko	15 janvier 2020	Frédéric Péters
-----------------------------------------------------------------	---------	-----------------	-----------------