

## w.c.s. - Bug #46854

### perf rendu /inspect (clés \_live\_)

21 septembre 2020 19:26 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	21 septembre 2020
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Frédéric Péters	<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>			
<a href="https://demarches-departement06.test.entrouvert.org/backoffice/management/inscription-aux-activites/687/inspect">https://demarches-departement06.test.entrouvert.org/backoffice/management/inscription-aux-activites/687/inspect</a> prend 22 secondes.			

### Révisions associées

#### Révision eff8a6cb - 22 septembre 2020 12:15 - Frédéric Péters

perf: cache live access data during request (#46854)

### Historique

#### #1 - 21 septembre 2020 19:30 - Frédéric Péters

A priori pas lié à cette nouvelle section (un for x in formdata.iter\_target\_datas(): print(x) de test est tout à fait rapide).

#### #2 - 21 septembre 2020 19:56 - Frédéric Péters

- *Sujet changé de perf rendu /inspect (parce que demandes et fiches liées ?) à perf rendu /inspect (clés \_live\_)*

- *Assigné à mis à Frédéric Péters*

En faisant

```
print('[%.2f] getting flat keys' % (time.time() - t0), file=fd)
keys = substvars.get_flat_keys()
print('[%.2f] getting flat keys -- done' % (time.time() - t0), file=fd)
print('[%.2f] iterating on flat keys' % (time.time() - t0), file=fd)
for k in sorted(keys):
    [...]
print('[%.2f] iterating on flat keys -- done' % (time.time() - t0), file=fd)
```

Ça donne :

```
[0.08] getting flat keys
[9.57] getting flat keys -- done
[9.57] iterating on flat keys
[22.12] iterating on flat keys -- done
```

Mon idée est maintenant de regarder dans les clés *live*, qui parfois s'enchainent (ex:

form\_var\_fiche\_senior\_live\_var\_conjoint\_senior\_live\_role\_editeur\_des\_quotas\_emails), sans elles, ça se passe ainsi :

```
[0.08] getting flat keys
[0.34] getting flat keys -- done
[0.34] iterating on flat keys
[0.62] iterating on flat keys -- done
```

(mesuré en retirant le "live" du inspect\_keys)

Vu que dans le live il y a chargement carddef + carddata,

```
carddef = CardDef.get_by_urlname(self._field.data_source['type'][8:])
carddata = carddef.data_class().get(self.raw)
```

ça n'est au final pas bien surprenant.

Il y aurait moyen de faire un get\_request().live\_card\_cache à l'image des datasources\_cache et wscalls\_cache, avec '%s-%s' % (self.\_field.data\_source['type'], self.raw) comme clé, pour voir.

### #3 - 21 septembre 2020 20:02 - Frédéric Péters

```
[0.07] getting flat keys
[0.77] getting flat keys -- done
[0.77] iterating on flat keys
[1.43] iterating on flat keys -- done
```

### #4 - 21 septembre 2020 20:03 - Frédéric Péters

- Fichier 0001-perf-cache-live-access-data-during-request-46854.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Ces mesures avec le patch attaché (toutes les mesures effectuées sur wcs.node1.test).

### #5 - 21 septembre 2020 20:49 - Frédéric Péters

- Fichier 0001-perf-cache-live-access-data-during-request-46854.patch ajouté

(adaptation test)

### #7 - 21 septembre 2020 23:50 - Thomas Noël

Je vois bien que cette partie du code fonctionne :

```
# cache during request
if not hasattr(request, 'live_card_cache'):
    request.live_card_cache = {}
cache_key = '%s-%s' % (self._field.data_source['type'], self.raw)
carddata = request.live_card_cache.get(cache_key)
if carddata is not None:
    if isinstance(carddata, Exception):
        raise carddata
    return LazyFormData(carddata)
```

mais peut-être que je suis trop vieux, j'aurais ajouté un else au cas où le cache n'est pas encore là, comme :

```
# cache during request
cache_key = '%s-%s' % (self._field.data_source['type'], self.raw)
if not hasattr(request, 'live_card_cache'):
    request.live_card_cache = {}
else:
    carddata = request.live_card_cache.get(cache_key)
    if carddata is not None:
        if isinstance(carddata, Exception):
            raise carddata
    return LazyFormData(carddata)
```

Ensuite concernant le test il y a un truc qui m'échappe : il fonctionne aussi très bien sans cache... Que teste-t-on exactement en demandant deux fois context['form\_var\_foo\_live\_name'] ?

### #8 - 22 septembre 2020 08:04 - Frédéric Péters

Ensuite concernant le test il y a un truc qui m'échappe : il fonctionne aussi très bien sans cache...

Oui, c'est qu'idéalement il y aurait mock/whatever et pouvoir vérifier qu'il n'y a pas d'appel, mais au final trop galère.

### #9 - 22 septembre 2020 08:56 - Frédéric Péters

Modification poussée dans la branche + aussi mock dans le test (pour juste la partie KeyError en fait ça va) pour voir que get\_by\_urlname n'est pas appelé la seconde fois.

### #10 - 22 septembre 2020 09:59 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

### #11 - 22 septembre 2020 12:15 - Frédéric Péters

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

commit eff8a6cb48d9826e62706bc497cd32ac92acbfef  
Author: Frédéric Péters <fpeters@entrouvert.com>  
Date: Mon Sep 21 20:01:41 2020 +0200

perf: cache live access data during request (#46854)

## #12 - 22 septembre 2020 19:16 - Frédéric Péters

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

### Fichiers

---

0001-perf-cache-live-access-data-during-request-46854.patch	1,68 ko	21 septembre 2020	Frédéric Péters
0001-perf-cache-live-access-data-during-request-46854.patch	2,45 ko	21 septembre 2020	Frédéric Péters