Passerelle - Development #17192

Ajouter une possibilité de cache à LoggedRequests

26 juin 2017 15:48 - Frédéric Péters

Statut:	Fermé	**Début:**	26 juin 2017
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

(similaire à ce qui existe dans combo)

Révisions associées

Révision 7fc51ef8 - 26 décembre 2017 13:24 - Frédéric Péters

utils: add cache support to requests wrapper (#17192)

Historique

#1 - 23 décembre 2017 13:10 - Frédéric Péters

- Fichier 0001-utils-add-cache-support-to-requests-wrapper-17192.patch ajouté

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

Par rapport à ce qui a été fait dans combo il n'y a par défaut pas de cache et les entêtes de la réponse sont également conservés.

#2 - 23 décembre 2017 16:37 - Benjamin Dauvergne

Il reste un print caché au milieu.

#3 - 23 décembre 2017 17:43 - Frédéric Péters

- Fichier 0001-utils-add-cache-support-to-requests-wrapper-17192.patch ajouté

oops.

#4 - 24 décembre 2017 21:20 - Thomas Noël

Pas très grave, mais on va lire le cache même si invalidate_cache est à True. Pour éviter ça, on pourrait ajouter un "not invalidate_cache" dans le if de la liste 201.

Ou mieux, on pourrait faire un cache.delete au lieu du cache.get si invalidate_cache est à True ; comme ça même si la requête plante ensuite, le cache a quand même été invalidé (et une requête suivante ne le prendre pas).

#5 - 25 décembre 2017 10:34 - Benjamin Dauvergne

Thomas Noël a écrit :

> Pas très grave, mais on va lire le cache même si invalidate_cache est à True. Pour éviter ça, on pourrait ajouter un "not invalidate_cache" dans le if de la liste 201.
>
> Ou mieux, on pourrait faire un cache.delete au lieu du cache.get si invalidate_cache est à True ; comme ça même si la requête plante ensuite, le cache a quand même été invalidé (et une requête suivante ne le prendre pas).

Ça dépend ce qu'on veut, ça peut-être bien que si la requête plante on garde quand même le cache même si idéalement on aurait voulu l'invalider, typiquement le invalidate cache va se faire sur un GET qui en fait est un POST (comme sur je sais plus quel code sur lequel bosse Josué en ce moment), et donc si l'opération échoue et bien on préfèrerait que le cache reste, mais bon ensuite on ne sait pas forcément ce que la condition "échouée" signifie exactement (ici on aura les échecs réseaux, mais si ça renvoie 500 on va quand même évincer le cache, des fois on pourra avoir du {'err': 1} comme erreur aussi qui nous rapportera que le truc derrière passerelle ne répond pas). Le cache c'est difficile, on couvrira pas la totalité des possibilités alors autant lister les cas d'usage qui nous semblent fréquent et ignorer le reste.

#6 - 26 décembre 2017 11:55 - Thomas Noël

> Le cache c'est difficile, on couvrira pas la totalité des possibilités alors autant lister les cas d'usage qui nous semblent fréquent et ignorer le reste.

En fait j'ai tiqué sur "invalidate_cache" qui n'invalide pas le cache, mais c'est le même code que #17056 où je trouvais le terme "très bien", alors je me tais et je acke :)

#7 - 26 décembre 2017 13:26 - Frédéric Péters

- Statut changé de En cours à Résolu (à déployer)

Poussé ainsi mais ça me va qu'on regarde pour modifier le comportement d'invalidate_cache (tant que les comportements de combo et passerelle sont similaires)

```
commit 7fc51ef806471329e06390cd1104e1bdd9a6bf66
Author: Frédéric Péters <fpeters@entrouvert.com>
Date:   Sat Dec 23 13:08:19 2017 +0100

    utils: add cache support to requests wrapper (#17192)
```

#8 - 04 août 2018 12:30 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-utils-add-cache-support-to-requests-wrapper-17192.patch	6,48 ko	23 décembre 2017	Frédéric Péters
0001-utils-add-cache-support-to-requests-wrapper-17192.patch	6,44 ko	23 décembre 2017	Frédéric Péters