

Passerelle - Development #24640

Cache pour les endpoint de connecteur

19 juin 2018 12:15 - Emmanuel Cazenave

Statut:	Fermé	Début:	19 juin 2018
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Permettre d'utiliser le framework de cache django sur l'endpoint d'un connecteur.			
Je pense un nouvel argument à <code>api/utils/endpoint.py::__init__.py</code> , disons <code>cached_settings</code> qui s'il est utilisé doit être une string par exemple <code>'MY_CONNECTOR_CACHE_DURATION'</code> .			
Ensuite à dans <code>views.GenericView.dispatch</code> on examine les settings django à la recherche <code>'MY_CONNECTOR_CACHE_DURATION'</code> et si ce settings est définis (et est un entier positif , etc), on utilise la méthode <code>cache_page</code> de django sur le <code>super(GenericEndpointView, self).dispatch</code> final.			

Révisions associées

Révision 86a91e41 - 02 juillet 2018 15:46 - Emmanuel Cazenave

add cache for connector endpoint (#24640)

Historique

#1 - 19 juin 2018 12:24 - Frédéric Péters

On pourrait commencer en simplement autorisant un paramètre de cache qui soit un entier ? (pour moi ça pourrait même s'arrêter là)

#2 - 19 juin 2018 12:58 - Benjamin Dauvergne

Oui sur BaseRessource ? Après pour les connecteurs spécifiques, style IWS, ça peut être intéressant de pouvoir donner une valeur par défaut qui ne soit pas 0 parce qu'on sait que ce sera toujours lent mais là en même temps comme la clé du cache dépendra de la requête je me dis que ce n'est de toute façon pas pertinent d'avoir quelque chose de global, et je retire tout ce que je dis.

#3 - 19 juin 2018 13:28 - Frédéric Péters

Je parlais d'un paramètre à `@endpoint()`, plutôt que permettre `cache_settings='MY_CONNECTOR_CACHE_DURATION'` permettre `cache_duration=300`.

#4 - 19 juin 2018 13:38 - Emmanuel Cazenave

Frédéric Péters a écrit :

On pourrait commencer en simplement autorisant un paramètre de cache qui soit un entier ? (pour moi ça pourrait même s'arrêter là)

Je suis partis sur un settings pour pouvoir facilement modifier les paramètres du cache dans les tests unitaires où à priori on ne voudra jamais de cache (et aussi pour ne pas avoir à faire une release de passerelle juste pour changer une durée de cache). Dans l'idée que dans les settings de tests on ne définira jamais de `MY_CONNECTOR_CACHE_DURATION` et voilà (et si vraiment on en veut, la fixture settings de pytest doit faire l'affaire).

Si on passe directement un entier à endpoint, comment on désactive le cache dans les tests ? Peut-être à coups de monkeypatch du `endpoint_info` ... ça s'annonce un peu pénible.

Alternativement peut-être normaliser le nom du settings : `"CACHE_NOM_DU_CONNECTEUR_NOM_DU_ENPOINT"`, et du coup ne même plus passer de paramètre à endpoint, juste aller examiner la présence du settings concerné dans `dispatch` ?

#5 - 19 juin 2018 13:48 - Thomas Noël

Je préfère vraiment un `cache_duration=300`, quitte à ce que le endpoint indique `cache_duration=self.cache_duration`, qui viendra de l'attribut du modèle (facile à modifier pour un test), ou même d'un `IntegerField` qui serait alors paramétrable dans l'UI lorsqu'on instancie le connecteur. Du grand n'importe quoi, oubliez-moi.

#6 - 19 juin 2018 14:04 - Emmanuel Cazenave

Emmanuel Cazenave a écrit :

Je suis partis sur un settings pour pouvoir facilement modifier les paramètres du cache dans les tests unitaires où à priori on ne voudra jamais de cache

Pour ça il y a en fait 'django.core.cache.backends.dummy.DummyCache' qui fait semblant mais qui ne cache rien.

Reste mon argument "ne pas faire une release de passerelle pour changer une durée de cache".

#7 - 19 juin 2018 14:42 - Benjamin Dauvergne

Je vais faire mon lourd mais que est le cas d'usage ? Les données quasi statiques (référentiels) ?

#8 - 19 juin 2018 14:48 - Emmanuel Cazenave

Mon cas d'usage c'est IWS : demandes de dates disponibles pour un rendez-vous.

Un endpoint de connecteur qui renvoie des données au format data source pour pouvoir être branché sur un champ liste.

Je ne connais pas le mécanisme de WCS, mais en tous cas il fait beaucoup d'appels pour une même demande.

Et donc non pas référentiel, parce que les dates disponibles dépendent en théorie de paramètres genre ville , adresse, type de demande, etc...

#9 - 19 juin 2018 15:26 - Benjamin Dauvergne

Donc tu fais des GET /iws/my-slug/get-dates/?ville=xxx&type_de_demande=12&guid=<form_number/form_uuid>&etc... et tu veux que ce soit mis en cache pour toujours pour que pour un même guid tu reçoives toujours le même token (ou plutôt que pour 2 guid différents tu ne reçoives jamais le même token).

La clé de cache ce serait un hash de request.get_full_path() ?

Je continue à penser que ça pourrait très bien aller dans le code de datasource pourvu qu'on utilise comme clé de cache l'URL de destination, et ça risque de resservir immédiatement (genre pour SOLIS je crois).

#10 - 19 juin 2018 15:27 - Benjamin Dauvergne

Pour être précis, je propose d'en faire une option de la définition de la datasource, cache_duration, bien sûr ça ne marchera pas pour les sources JSONP jusqu'à ce qu'on ait un proxy dans w.c.s.

#11 - 19 juin 2018 15:52 - Emmanuel Cazenave

Benjamin Dauvergne a écrit :

Donc tu fais des GET /iws/my-slug/get-dates/?ville=xxx&type_de_demande=12&guid=<form_number/form_uuid>&etc... et tu veux que ce soit mis en cache pour toujours pour que pour un même guid tu reçoives toujours le même token (ou plutôt que pour 2 guid différents tu ne reçoives jamais le même token).

Je pensais à 5 min de cache plutôt que "pour toujours", le temps de remplir une demande quoi, juste pour pas retaper dans le webservice métier quand on valide la page de formulaire, puis quand on soumet la demande.

La clé de cache ce serait un hash de request.get_full_path() ?

Dans ce que je propose c'est le cache_page de django qui crée la clé, mais il fait un truc dans ce gout là oui.

Je continue à penser que ça pourrait très bien aller dans le code de datasource pourvu qu'on utilise comme clé de cache l'URL de destination, et ça risque de resservir immédiatement (genre pour SOLIS je crois).

Je connais absolument pas ce code, je me déclare incompetent.

#12 - 20 juin 2018 12:32 - Emmanuel Cazenave

- Fichier 0001-add-cache-for-connector-endpoint-24640.patch ajouté

- Patch proposed changé de Non à Oui

J'ai cru que je pourrais m'en sortir en deux coups de cuillères à pot mais non pas du tout.

Le patch naïf ci-joint se fait complètement avoir par les timestamp et signature dans les query string : utilisés par django pour calculer la clé de cache, résultat ça cache walou.

#13 - 20 juin 2018 14:22 - Emmanuel Cazenave

Et donc pas moyen d'accéder à la façon dont est générée la clé de cache, il faudrait donc redéfinir un truc à la `django.views.decorators.cache.cache_page` qui lui même utilise `django.middleware.cache.CacheMiddleware` à redéfinir aussi, c'est à ce niveau qu'on accède à la génération de la clé.

Bref assez lourdingue, pour faire quasiment la même chose à très peu de choses près.

Domage que timestamp et signature soient dans la query string, dans des headers on aurait été bons.

#14 - 20 juin 2018 14:31 - Emmanuel Cazenave

- Patch proposed changé de Oui à Non

#15 - 20 juin 2018 14:37 - Frédéric Péters

Plutôt qu'essayer de profiter d'un décorateur de cache de django, gérer ça "manuellement" dans le perform de `GenericEndpointView` ?

Genre vaguement cette trame :

```
-         return self.endpoint(request, **self.get_params(request, *args, **kwargs))
+         params = self.get_params(request, *args, **kwargs)
+         if method == 'GET' and self.endpoint.cache_duration:
+             cache_key = hashlib.md5(repr(self.endpoint) + repr(params)).hexdigest() # me semble suffisant, à
vérifier
+             if ... has cache: return ...
+             result = self.endpoint(request, params)
+             if method == 'GET' and self.endpoint.cache_duration:
+                 pass # mise en cache
+             return result
```

#16 - 20 juin 2018 19:27 - Emmanuel Cazenave

- Fichier `0001-add-cache-for-connector-endpoint-24640.patch` ajouté

- Patch proposed changé de Non à Oui

Frédéric Péters a écrit :

Plutôt qu'essayer de profiter d'un décorateur de cache de django, gérer ça "manuellement" dans le perform de `GenericEndpointView` ?

Yes indeed.

J'ai ajouté `self.get_object().slug` dans la clé (pour le cas plusieurs instances d'un même connecteur), et laissé de côté les tests sur `request.GET`, `perform` étant appelé uniquement dans `get`.

#17 - 20 juin 2018 20:03 - Frédéric Péters

et laissé de côté les tests sur `request.GET`, `perform` étant appelé uniquement dans `get`.

Sauf que :

```
def post(self, request, *args, **kwargs):
    return self.get(request, *args, **kwargs)
```

#18 - 20 juin 2018 20:09 - Frédéric Péters

Côté tests, je préférerais ne pas toucher à `passerelle/contrib/stub_invoices/models.py`, cette méthode de test pourrait être ajoutée par monkeypatch dans le test ?

#19 - 21 juin 2018 12:53 - Emmanuel Cazenave

- Fichier `0001-add-cache-for-connector-endpoint-24640.patch` ajouté

```
def post(self, request, *args, **kwargs):
    return self.get(request, *args, **kwargs)
```

So sweet.

Voilà, avec tous les changements demandés.

Test étendu pour vérifier que la cache n'est pas touché dans un POST.

Et je ne touche plus à CACHES dans les settings de test : ce n'était en fait pas nécessaire ici, anticipation du besoin à venir pour les tests sur des endpoint avec du cache. Sauf que ça cassait d'autres tests qui ont besoin d'un vrai cache (ex: test_requests.py).

La fixture endpoint_dummy_cache incluse dans ce patch devrait couvrir ce besoin.

#20 - 02 juillet 2018 12:27 - Benjamin Dauvergne

- Statut changé de Nouveau à Solution validée

Ack.

#21 - 02 juillet 2018 15:47 - Emmanuel Cazenave

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

```
commit 86a91e41e159f5845c7837175f52dbff6d745ef8
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date: Wed Jun 20 10:56:22 2018 +0200
```

```
add cache for connector endpoint (#24640)
```

#22 - 04 août 2018 12:29 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-add-cache-for-connector-endpoint-24640.patch	4,21 ko	20 juin 2018	Emmanuel Cazenave
0001-add-cache-for-connector-endpoint-24640.patch	6,44 ko	20 juin 2018	Emmanuel Cazenave
0001-add-cache-for-connector-endpoint-24640.patch	6,64 ko	21 juin 2018	Emmanuel Cazenave