Passerelle - Development #27654

requests : permettre de garder les cookies

29 octobre 2018 18:13 - Emmanuel Cazenave

Statut:	Fermé	**Début:**	29 octobre 2018
Priorité:	Normal	**Echéance:**	
Assigné à:	Emmanuel Cazenave	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

Plus de cookies depuis #24619 mais dans #27653, la méthode d'authentification passe par un cookie.

Donc une option pour pouvoir les garder.

Demandes liées:

Lié à Passerelle - Bug #73655: test test_endpoint_cookies qui interroge vraim...	**Fermé**	**20 janvier 2023**

Révisions associées

Révision 58b5415b - 06 novembre 2018 09:51 - Emmanuel Cazenave

allow cookies usage in endpoint requests (#27654)

Historique

#1 - 30 octobre 2018 12:25 - Emmanuel Cazenave

- Fichier 0001-utils-allow-cookies-usage-on-requests-27654.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

#2 - 30 octobre 2018 13:18 - Frédéric Péters

On va se retrouver sur le soucis de cookies partagés entre différentes requêtes alors que le souhait est uniquement d'avoir ça le long de la requête, il me semble; il faudrait plutôt voir pour qu'en début de endpoint un CookieJar dédié soit assigné, libéré à la fin.

#3 - 30 octobre 2018 14:00 - Emmanuel Cazenave

Pas sûr de bien te suivre, je détaille mon cas d'usage.
Pour une requête arrivant à un endpoint du connecteur, je dois faire les requêtes suivantes vers planitech :

- première requête pour obtenir un token.
- une deuxième requête pour envoyer un hash de ce token + password. La réponse à cette requête positionne un cookie qui permet l'authentification pour les requêtes suivantes.
- troisième requête : je fais vraiment quelque chose avec l'API de planitech

Et peut-être une une quatrième requête comme la troisième selon le endpoint, je ne sais pas encore, à voir.

#4 - 30 octobre 2018 14:24 - Emmanuel Cazenave

Bon je te relis mieux et je te comprends mieux, maintenant je réfléchis.

#5 - 30 octobre 2018 14:30 - Frédéric Péters

Bon, avant tout, l'origine de la suppression des cookies c'est une situation dans combo, qui peut-être ne s'appliquait pas à Passerelle. Et à découvrir ça, il me semble aussi que ce patch ne marcherait pas pour toi.

Sur l'idée, dans le connecteur, qu'on trouve des self.requests.get(...), mais ce self.requests, c'est :

```
    @property
    def requests(self):
        return passerelle.utils.Request(resource=self, logger=self.logger)
```

et donc, créé à chaque fois, ce qui fait cookiejar vide à chaque appel.

Ton test passe à côté de ça, en créant et en réutilisant un même objet request.

Ce que cela voudrait dire, c'est que le clear_cookies() n'était pas utile, mais aussi, que les cookies sont de toute façon absent et qu'il te faut trouver autre chose.

Et donc les choses faisant leur tour, pas le sens que j'imaginais mais quand même, ma suggestion du commentaire précédent se traduirait par quelque chose de ce genre :

```
diff --git a/passerelle/utils/__init__.py b/passerelle/utils/__init__.py
index 815fe79..604d06d 100644
--- a/passerelle/utils/__init__.py
+++ b/passerelle/utils/__init__.py
@@ -214,8 +214,9 @@ class Request(RequestSession):
         if 'timeout' not in kwargs:
             kwargs['timeout'] = settings.REQUESTS_TIMEOUT

-        # don't use persistent cookies
-        self.cookies.clear()
+        # persist cookie for the endpoint duration
+        if self.resource and hasattr(self.resource, 'cookiejar'):
+            self.cookies = self.resource.cookiejar

         response = super(Request, self).request(method, url, **kwargs)

diff --git a/passerelle/views.py b/passerelle/views.py
index 33b8ccc..3ebc229 100644
--- a/passerelle/views.py
+++ b/passerelle/views.py
@@ -365,6 +365,7 @@ class GenericEndpointView(GenericConnectorMixin, SingleObjectMixin, View):
             if result is not None:
                 return result

+        self.cookiejar = requests.cookies.cookiejar_from_dict({})
         result = self.endpoint(request, **params)
         if request.method == 'GET' and self.endpoint.endpoint_info.cache_duration:
             cache.set(cache_key, result, self.endpoint.endpoint_info.cache_duration)
```

#6 - 30 octobre 2018 16:51 - Emmanuel Cazenave

> Ce que cela voudrait dire, c'est que le clear_cookies() n'était pas utile, mais aussi, que les cookies sont de toute façon absent et qu'il te faut trouver autre chose.

clear_cookies clairement pas utile.
Sur #24404 dans combo la session est partagée :

```
diff --git a/combo/utils/requests_wrapper.py b/combo/utils/requests_wrapper.py
index 6dbde7d..52594b9 100644
--- a/combo/utils/requests_wrapper.py
+++ b/combo/utils/requests_wrapper.py
@@ -128,4 +128,5 @@ class Requests(RequestsSession):

         return response

+
 requests = Requests()
```

Ici le fait que l'attribut requests soit une factory de session nous prémunit des problèmes de #24404.

En tenant compte de cette histoire de factory, mon patch marche marche bel et bien en procédant comme suit :

```
def _login(self):
        auth_url = urlparse.urljoin(self.url, 'auth')
        session = self.requests
        ...
        # faire ce qui doit être fait avec cette session pour obtenir un cookie
        .....
        return session

    @endpoint(methods=['get'], perm='can_access')
    def getplaceslist(self, request):
        session = self._login()
        response = session.get(urlparse.urljoin(self.url, 'getPlacesList'))
        data = mste.decode(response.json())
        return {'data': data}
```

Je comprends l'idée ton patch, mais je crois que ça marchera pas parce que dans passerelle/views.py::perform on pas accès à l'objet resource sur lequel il faudrait positionner le cookiejar.

Bref je pense que le bon patch c'est ça (testé en procédant comme décrit précédemment) :

```
diff --git a/passerelle/utils/__init__.py b/passerelle/utils/__init__.py
index 815fe79..766dd78 100644
--- a/passerelle/utils/__init__.py
+++ b/passerelle/utils/__init__.py
@@ -214,9 +214,6 @@ class Request(RequestSession):
         if 'timeout' not in kwargs:
             kwargs['timeout'] = settings.REQUESTS_TIMEOUT

-        # don't use persistent cookies
-        self.cookies.clear()
```

Je veux des cookies, j'utilise la même session. J'en veux pas j'en demande une nouvelle via l'attribut requests.
Non ?

#7 - 30 octobre 2018 17:00 - Frédéric Péters

> En tenant compte de cette histoire de factory, mon patch marche marche bel et bien en procédant comme suit :

Mais c'est moche comme tout, ça suppose se souvenir que self.requests c'est pas juste un raccourci qui va bien (logging etc.) mais quelque chose de plus compliqué, avec un état, qu'on doit appeler mais conserver si on veut, etc.

> Je comprends l'idée ton patch, mais je crois que ça marchera pas parce que dans passerelle/views.py::perform on pas accès à l'objet resource sur lequel il faudrait positionner le cookiejar.

Il fallait bien sûr y lire connector.cookiejar = ... et pas self.cookiejar = ... parce que oui, "resource" c'est "connector".

#8 - 30 octobre 2018 17:44 - Emmanuel Cazenave

Frédéric Péters a écrit :

> > En tenant compte de cette histoire de factory, mon patch marche marche bel et bien en procédant comme suit :
>
> Mais c'est moche comme tout, ça suppose se souvenir que self.requests c'est pas juste un raccourci qui va bien (logging etc.) mais quelque chose de plus compliqué, avec un état, qu'on doit appeler mais conserver si on veut, etc.

J'ai l'opinion inverse : je trouve que c'est moche comme tout d'essayer de re-implémenter nous même une notion de session dans un endpoint alors qu'on l'a déjà gratos et proprement dans une session requests qui est faite exactement pour ça.

#9 - 30 octobre 2018 17:51 - Frédéric Péters

L'attribut s'appelle requests, c'est une propriété, ce qui ne laisse pas non plus penser qu'à chaque appel il y aura un résultat différent.

D'un point de vue documentation, c'est simple et clair d'écrire « self.requests utilisé ici est un wrapper léger au-dessus du module Requests, qui ajoute une journalisation et une expiration automatique des appels. », j'aimerais pouvoir conserver ça.

Ma proposition le permet, tout en gagnant "les cookies persistent le long des différentes requêtes que peut faire l'endpoint".

#10 - 30 octobre 2018 19:31 - Emmanuel Cazenave

- Fichier 0001-wip.patch ajouté

Et un truc dans ce goût là ça t'irait ?

Ça aurait le mérite de nous éviter d'avoir à implémenter notre gestion des cookies, plus tard notre cookies.clear() quand on tombera sur un cas tordu qui le nécessitera, etc.

Testé vite fait dans un shell, ça a l'air ok du point de vu de l'isolation des sessions :

```
models.CsvDataSource.objects.all()
Out[3]: <InheritanceQuerySet [<CsvDataSource: Réservation salle- Liste des salles>]>
In [4]: first = models.CsvDataSource.objects.all()[0]
In [5]: first
Out[5]: <CsvDataSource: Réservation salle- Liste des salles>
In [7]: first._toto = 'tata'
```

```
In [8]: second = models.CsvDataSource.objects.all()[0]
In [9]: second._toto
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-9-3af6b2cc2d39> in <module>()
----> 1 second._toto

AttributeError: 'CsvDataSource' object has no attribute '_toto'
<pre>
```

#11 - 30 octobre 2018 19:39 - Frédéric Péters

À expliquer c'est "Si jamais vous avez besoin de persister les cookies le long des appels de votre endpoint, remplacez vos self.request. par des self.session." ? Je ne trouve pas ça terrible et je continue à penser que ma proposition fait le job sans rien demander à la documentation.

#12 - 01 novembre 2018 13:37 - Emmanuel Cazenave

Avec quelques aménagements nécessaires.

#13 - 01 novembre 2018 13:37 - Emmanuel Cazenave

- Fichier 0001-allow-cookies-usage-in-endpoint-requests-27654.patch ajouté

#14 - 02 novembre 2018 10:05 - Frédéric Péters

Je suis surpris par la nécessité de :

```
        if getattr(self, 'cookies', None) and self.resource \
           and hasattr(self.resource, 'cookiejar'):
            self.resource.cookiejar = self.cookies
```

j'aurais imaginé requests utilisé le cookiejar qu'on lui donnait, pas l'écraser. Je vais lire un peu.

#15 - 02 novembre 2018 10:25 - Frédéric Péters

> j'aurais imaginé requests utilisé le cookiejar qu'on lui donnait, pas l'écraser. Je vais lire un peu.

Et c'est donc httmock qui remplace l'attribut. Ces trois lignes sont uniquement utiles pour httmock. Ça m'ennuie pas mal mais je ne vais pas me battre là... Je propose juste d'être explicite sur les raisons, façon :

```
--- a/passerelle/utils/__init__.py
+++ b/passerelle/utils/__init__.py
@@ -214,14 +214,16 @@ class Request(RequestSession):
         if 'timeout' not in kwargs:
             kwargs['timeout'] = settings.REQUESTS_TIMEOUT

-        # persist cookie for the endpoint duration
         if self.resource and hasattr(self.resource, 'cookiejar'):
+            # use cookies that will last the whole endpoint duration
             self.cookies = self.resource.cookiejar

         response = super(Request, self).request(method, url, **kwargs)

-        if getattr(self, 'cookies', None) and self.resource \
-           and hasattr(self.resource, 'cookiejar'):
+        if self.resource and hasattr(self.resource, 'cookiejar'):
+            # make sure we get the new cookiejar; it is actually not necessary
+            # in normal operations but httmock will replace the cookiejar
+            # instead of filling it :/
             self.resource.cookiejar = self.cookies

         if method == 'GET' and cache_duration and (response.status_code // 100 == 2):
```

#16 - 02 novembre 2018 10:30 - Emmanuel Cazenave

Frédéric Péters a écrit :

> Et c'est donc httmock qui remplace l'attribut. Ces trois lignes sont uniquement utiles pour httmock. Ça m'ennuie pas mal mais je ne vais pas me battre là...

Merci, je pensais que c'était requests, sans avoir pris le temps de vérifier.

Je trouve ça naze aussi, je prends un moment pour voir si je peux pas tester autrement et virer ces lignes.

#17 - 02 novembre 2018 14:17 - Emmanuel Cazenave

- Fichier 0001-allow-cookies-usage-in-endpoint-requests-27654.patch ajouté

Essayé de tester avec requests-mock mais pareil que httmock.

Le getattr(self, 'cookies', None) est bien nécessaire (toujours pour les tests unitaires), bien moche, pas d'autre idée.

#18 - 02 novembre 2018 14:28 - Frédéric Péters

> getattr(self, 'cookies', None)

J'avais dégagé ça de mon diff, je ne pense pas que c'est nécessaire. (et ce que je trouve particulièrement moche c'est le \ de fin de ligne)

#19 - 02 novembre 2018 14:39 - Emmanuel Cazenave

- Fichier 0001-allow-cookies-usage-in-endpoint-requests-27654.patch ajouté

#20 - 02 novembre 2018 17:58 - Emmanuel Cazenave

- Fichier 0001-allow-cookies-usage-in-endpoint-requests-27654.patch ajouté

Sans ce qui nous ennuie.

En contrepartie, j'utilise httpbin pour les tests, qui lance en vrai serveur http en local (j'ai trouvé ça dans les tests unitaires de requests).

#21 - 06 novembre 2018 08:49 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

Yes, super.

#22 - 06 novembre 2018 10:01 - Emmanuel Cazenave

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

```
commit 58b5415b879f3250f22ed6f67b9c59cce7061042
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date:   Thu Nov 1 13:18:20 2018 +0100

    allow cookies usage in endpoint requests (#27654)
```

#23 - 30 novembre 2018 12:20 - Emmanuel Cazenave

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

#24 - 03 décembre 2018 15:15 - Benjamin Dauvergne

- Statut changé de Solution déployée à Fermé

#25 - 17 avril 2023 17:53 - Emmanuel Cazenave

- Lié à Bug #73655: test test_endpoint_cookies qui interroge vraiment httpbin.org ajouté

Fichiers

0001-utils-allow-cookies-usage-on-requests-27654.patch	1,98 ko	30 octobre 2018	Emmanuel Cazenave
0001-wip.patch	1,49 ko	30 octobre 2018	Emmanuel Cazenave
0001-allow-cookies-usage-in-endpoint-requests-27654.patch	6,29 ko	01 novembre 2018	Emmanuel Cazenave
0001-allow-cookies-usage-in-endpoint-requests-27654.patch	6,5 ko	02 novembre 2018	Emmanuel Cazenave
0001-allow-cookies-usage-in-endpoint-requests-27654.patch	6,47 ko	02 novembre 2018	Emmanuel Cazenave
0001-allow-cookies-usage-in-endpoint-requests-27654.patch	5,07 ko	02 novembre 2018	Emmanuel Cazenave