

w.c.s. - Development #68337

Exploitation de APIClient

22 août 2022 17:03 - Emmanuel Cazenave

Statut:	Fermé	Début:	22 août 2022
Priorité:	Normal	Echéance:	
Assigné à:	Emmanuel Cazenave	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Les accès donnés dans authentic doivent fonctionner ici.			
Demandes liées:			
Lié à Authentic 2 - Development #66985: gestion centralisée des accès aux API...		Fermé	05 juillet 2022
Bloqué par w.c.s. - Development #68527: Utiliser la lib responses dans les tests		Fermé	29 août 2022

Révisions associées

Révision 1214963e - 12 septembre 2022 17:27 - Emmanuel Cazenave

api: extend api accesess with idp (#68337)

Historique

#1 - 22 août 2022 17:03 - Emmanuel Cazenave

- Lié à Development #66985: gestion centralisée des accès aux API / Infrastructure minimale pour accès HTTP basic ajouté

#2 - 23 août 2022 16:58 - Emmanuel Cazenave

- Statut changé de Nouveau à En cours

#3 - 24 août 2022 18:24 - Emmanuel Cazenave

- Fichier 0001-api-add-api-client-authentication-68337.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

C'est tout simple.

Le plus dur les tests : j'étais parti sur de grande ambitions de factorisation (via des fixture) des tests existants sur l'authentification de l'API, dans l'idée qu'ensuite je n'aurais qu'à ajouter quelques lignes de code là dedans, pour qu'une multitude de tests tournent sur l'authentification via api client.

Mais marche trop haute, tout à la poubelle, et écriture de tests a part.

#4 - 27 août 2022 20:44 - Frédéric Péters

Il y a déjà deux formes pour le mock de réponse HTTP dans les tests de wcs : la fixture http_requests pour des réponses "communes" et le mock.patch('wcs.qommon.misc.urlopen') pour des réponses ad hoc, utilisé par exemple ici :

```
with mock.patch('wcs.qommon.misc.urlopen') as urlopen:
    value = [{'id': '1', 'text': 'foo'}]
    urlopen.side_effect = lambda *args: io.StringIO(json.dumps({'data': value, 'err': 0}))
    assert datasource.get_structured_value('1') == value[0]
    assert urlopen.call_count == 1
```

J'aimerais ne pas voir l'introduction d'une troisième méthode. (le module "responses")

..

```
+         if hasattr(self.user, 'restrict_to_anonymised_data'):
+             return self.user.restrict_to_anonymised_data
```

c'est moche (hasattr en général), je préférerais la modif (pas testée)

```
@@ -64,6 +64,7 @@ class ApiAccess(XmlStorableObject):
```

```

def __init__(self, api_access):
    self.api_access = api_access
+     self.restrict_to_anonymised_data = api_access.restrict_to_anonymised_data

```

Plus fondamentalement, je suis un peu gêné par cette introduction APIClientUser : nouvel objet, un peu pareil un peu différent, pour lequel ma première réaction juste au-dessus est d'essayer de gommer les différences.

Ça n'aurait pas pu être géré dans ApiAccess, que le ApiAccess.get_with_credentials() se termine si rien trouvé par l'appel get_api_client et crée un objet ApiAccess avec les infos obtenues (sans l'enregistrer) ? (oui ça demande d'aussi gérer has_anonymised_data_api_restriction).

#5 - 29 août 2022 14:18 - Benjamin Dauvergne

- Statut changé de Solution proposée à Information nécessaire

Je ne relis pas encore, j'attendrai une réponse aux questions de Fred.

#6 - 29 août 2022 14:45 - Frédéric Péters

(pour info mon commentaire était une relecture)

#7 - 29 août 2022 15:45 - Emmanuel Cazenave

- Fichier 0001-api-extend-api-accesses-with-idp-68337.patch ajouté

- Statut changé de Information nécessaire à Solution proposée

Frédéric Péters a écrit :

J'aimerais ne pas voir l'introduction d'une troisième méthode. (le module "responses")

On peut négocier ?

Je trouve qu'il y a vraiment pas photo en termes de facilité d'usage et de lisibilité. Ci-dessous le premier tests avec et sans responses pour comparaison :

@responses.activate

```

def test_idp_http_error(app, pub, role):
    # No authentication : 403
    app.get('/api/forms/', status=403)
    app.set_authorization(('Basic', ('test-client', '12345')))
    # check-api-client returns 403
    responses.add(
        responses.POST,
        'https://authentic.example.invalid/api/check-api-client/',
        status=403,
    )
    app.get('/api/forms/', status=403)
    # check api call to authentic
    assert len(responses.calls) == 1
    request = responses.calls[0].request
    assert request.headers['Accept'] == 'application/json'
    assert request.headers['Content-Type'] == 'application/json'
    assert json.loads(request.body) == {'identifiant': 'test-client', 'password': '12345'}

```

```

def test_idp_http_error(app, pub, role):
    # No authentication : 403
    app.get('/api/forms/', status=403)
    app.set_authorization(('Basic', ('test-client', '12345')))
    # check-api-client returns 403
    with mock.patch('wcs.qommon.misc._http_request') as http_request:
        http_request.return_value = None, 403, None, None
        app.get('/api/forms/', status=403)
        # check api call to authentic
        assert http_request.call_count == 1
        assert http_request.call_args[0][0].startswith(
            'https://authentic.example.invalid/api/check-api-client/'
        )
        assert http_request.call_args[0][1] == 'POST'
        assert http_request.call_args[1]['headers']['Accept'] == 'application/json'
        assert http_request.call_args[1]['headers']['Content-type'] == 'application/json'
        assert json.loads(http_request.call_args[0][2]) == {'identifiant': 'test-client', 'password': '12345'}

```

Pour info c'est déjà introduit dans passerelle.

Plus fondamentalement, je suis un peu gêné par cette introduction APIClientUser

Voilà.

(oui ça demande d'aussi gérer has_anonymised_data_api_restriction)

Pas bien compris, tu vois des ajustements en plus à faire par rapport à cette deuxième version du patch ?

#8 - 29 août 2022 16:07 - Frédéric Péters

On peut négocier ?

Remplacer tous les mock/urlopen par ça et idéalement aussi la fixture http_requests; l'idée étant vraiment de ne pas ajouter une troisième méthode.

Pour info c'est déjà introduit dans passerelle.

Oui j'avais vu. (mais ça a été introduit avec une justification (assert quelque chose) qui ne tient pas ici).

..

Concernant le nouveau commit,

```
+ class InnerRole:
```

Plutôt non; peut-être quelque chose façon

```
@@ -75,7 +77,7 @@ class ApiAccess(XmlStorableObject):
    return self.roles

    user = RestrictedApiUser(self)
-    user.roles = [x.id for x in self.get_roles()]
+    user.roles = self.get_role_ids()
    return user

    @classmethod
@@ -85,6 +87,12 @@ class ApiAccess(XmlStorableObject):
    raise KeyError
    return api_access.get_as_api_user()

+    def get_role_ids(self):
+        if self._role_ids is not Ellipsis:
+            return self._role_ids
+        self._role_ids = [x.id for x in self.get_roles()]
+        return self._role_ids
+
```

et dans le get_from_idp() faire

```
api_access._roles_ids = data['data']['roles']

+ from .common.errors import ConnectionError
+ from .common.misc import http_post_request
```

ceux-ci peuvent être importés en haut du module.

(oui ça demande d'aussi gérer has_anonymised_data_api_restriction)

Pas bien compris, tu vois des ajustements en plus à faire par rapport à cette deuxième version du patch ?

C'est que dans le patch précédent il y avait aussi une modification à cette méthode (has_anonymised_data_api_restriction) qu'il me semblait éventuellement devoir reprendre, parce qu'elle fait :

```
if orig:
    api_access = ApiAccess.get_by_identifiant(orig)
```

et que ça va retourner None ici; mais comme il n'y aura en fait pas de paramètre orig ça va passer.

#9 - 29 août 2022 18:06 - Emmanuel Cazenave

- Fichier 0001-api-extend-api-accesses-with-idp-68337.patch ajouté
- Statut changé de Solution proposée à Nouveau
- Assigné à Emmanuel Cazenave supprimé

#11 - 29 août 2022 18:15 - Frédéric Péters

- Patch proposed changé de Oui à Non

(mais ça a été introduit avec une justification (assert quelque chose) qui ne tient pas ici).

J'ai tenté de respecter le dogme que je ne partage pas, (...)

Le "dogme" c'est ne pas multiplier les manières de faire, je pense ça critique pour garder quelque chose de maintenable et accessible à tout le monde.

J'entends que tu ne penses pas ça et aussi qu'aller modifier des dizaines d'occurrence de `with mock.patch('wcs.qommon.misc.urlopen')` n'est pas super fun.

#12 - 29 août 2022 18:29 - Emmanuel Cazenave

- Statut changé de Nouveau à En cours
- Assigné à mis à Emmanuel Cazenave

#13 - 01 septembre 2022 17:05 - Emmanuel Cazenave

- Bloqué par Development #68527: Utiliser la lib responses dans les tests ajouté

#14 - 01 septembre 2022 17:27 - Emmanuel Cazenave

- Fichier 0001-api-extend-api-accesses-with-idp-68337.patch ajouté
- Statut changé de En cours à Solution proposée
- Patch proposed changé de Non à Oui

Avec l'utilisation de `with responses.RequestsMock()` as `rsps: plut@t` que `@responses.activate` pour être raccord avec la forme commune.

#15 - 12 septembre 2022 14:00 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

Ok, une fois qu'on a le tout correct (ui côté authentic etc.) il faudra aussi un ticket pour ne plus exposer la création d'accès API via wcs, qu'une seule méthode soit exposée.

#16 - 12 septembre 2022 18:30 - Emmanuel Cazenave

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

```
commit 1214963e728d9387488e0cfb82aef69181a20084
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date: Mon Aug 29 14:22:08 2022 +0200
```

```
api: extend api accesses with idp (#68337)
```

#17 - 12 septembre 2022 18:33 - Emmanuel Cazenave

Frédéric Péters a écrit :

Ok, une fois qu'on a le tout correct (ui côté authentic etc.)

Je me disais qu'on attendrait quelques semaines avant de cacher les "anciens" accès, le temps pour des gens de se servir de ces nouveaux accès et de les éprouver.

#18 - 15 septembre 2022 01:14 - Transition automatique

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

#20 - 20 novembre 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-api-add-api-client-authentication-68337.patch	14,3 ko	24 août 2022	Emmanuel Cazenave
0001-api-extend-api-accesses-with-idp-68337.patch	12,8 ko	29 août 2022	Emmanuel Cazenave
0001-api-extend-api-accesses-with-idp-68337.patch	13,6 ko	29 août 2022	Emmanuel Cazenave
0001-api-extend-api-accesses-with-idp-68337.patch	13,2 ko	01 septembre 2022	Emmanuel Cazenave