

Combo - Development #13125

faire que utils.requests puisse trouver tout seul le service distant dans KNOWN_SERVICES

08 septembre 2016 17:53 - Thomas Noël

Statut:	Fermé	Début:	08 septembre 2016
Priorité:	Normal	Echéance:	
Assigné à:	Thomas Noël	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
actuellement il faut envoyer à combo.utils.requests un remote_service à aller chercher dans KNOWN_SERVICES			
je propose que si on envoie "remote_service='auto'" alors requests se charge de chercher le service tout seul en fonction du nom dans l'URL à interroger			

Révisions associées

Révision e9e2f34e - 21 septembre 2016 13:19 - Thomas Noël

utils.requests: remote_service can be guessed (#13125)

Historique

#1 - 08 septembre 2016 17:55 - Thomas Noël

un début de patch avant que je rentre chez moi, pour percevoir l'idée (et me crier dessus si elle est trop mauvaise)

```
diff --git a/combo/utils.py b/combo/utils.py
index 8f22ece..700d059 100644
--- a/combo/utils.py
+++ b/combo/utils.py
@@ -87,6 +87,21 @@ class Requests(RequestsSession):
     raise_if_not_cached = kwargs.pop('raise_if_not_cached', False)
     log_errors = kwargs.pop('log_errors', True)

+     if remote_service == 'auto':
+         scheme, netloc, old_path, params, old_query, fragment = urlparse.urlparse(url)
+         for services in settings.KNOWN_SERVICES.values():
+             for service in services.values():
+                 remote_url = service.get('url')
+                 remote_netloc = urlparse.urlparse(remote_url).netloc.split(':')[0]
+                 if remote_netloc == netloc:
+                     remote_service = service
+                     break
+             else:
+                 continue
+         break
+     if remote_service:
+         url = urlparse.urlunparse((' ', ' ', path, params, query, fragment))

+     if remote_service:
+         if isinstance(user, dict):
+             query_params = user.copy()
```

note: on pourrait décider de logger+planter si aucun service n'est trouvé

#2 - 08 septembre 2016 17:56 - Thomas Noël

- Statut changé de Nouveau à En cours

#3 - 08 septembre 2016 18:23 - Benjamin Dauvergne

Oui je pense qu'on peut décider de logger une erreur, bien décrite.

#4 - 09 septembre 2016 16:06 - Thomas Noël

- Fichier 0001-utils.requests-remote_service-can-be-guessed-13125.patch ajouté

- Patch proposed changé de Non à Oui

#5 - 09 septembre 2016 16:11 - Thomas Noël

- Tracker changé de Bug à Development

#6 - 09 septembre 2016 16:30 - Benjamin Dauvergne

Tu sépares bien le port du netloc venant de KNOWN_SERVICES mais pas celui venant de l'URL passée à requests, il faut le faire dans les deux cas ou ne pas le faire, mais il faut choisir, je pencherai pour ne pas le faire et j'irai jusqu'à comparer le scheme aussi, port != ou scheme != pour moi ce n'est pas le même service.

Sinon pour le reste ça me va. Mais en fait je ne comprends pas trop le code de la classe Requests ça fait des trucs bizarre au lieu de simplement ajouter orig et la signature à l'URL ça va mélanger service[url] de dedans, je suppose qu'il y a une raison mais ce serait bien de la mettre dans le code à un jour.

```
if remote_service:
    if isinstance(user, dict):
        query_params = user.copy()
    elif not user or not user.is_authenticated():
        if without_user:
            query_params = {}
        else:
            query_params = {'NameID': '', 'email': ''}
    else:
        query_params = {}
        if federation_key == 'nameid':
            query_params['NameID'] = user.saml_identifiers.first().name_id
        elif federation_key == 'email':
            query_params['email'] = user.email
        else: # 'auto'
            if hasattr(user, 'saml_identifiers') and user.saml_identifiers.exists():
                query_params['NameID'] = user.saml_identifiers.first().name_id
            else:
                query_params['email'] = user.email

query_params['orig'] = remote_service.get('orig')

remote_service_base_url = remote_service.get('url')
scheme, netloc, old_path, params, old_query, fragment = urlparse.urlparse(
    remote_service_base_url)

query = urlencode(query_params)
if '?' in url:
    path, old_query = url.split('?')
    query += '&' + old_query
else:
    path = url

url = urlparse.urlunparse((scheme, netloc, path, params, query, fragment))
```

#7 - 09 septembre 2016 17:02 - Thomas Noël

- Fichier 0001-utils.requests-remote_service-can-be-guessed-13125.patch ajouté

Benjamin Dauvergne a écrit :

Tu sépares bien le port du netloc venant de KNOWN_SERVICES mais pas celui venant de l'URL passée à requests, il faut le faire dans les deux cas ou ne pas le faire, mais il faut choisir, je pencherai pour ne pas le faire et j'irai jusqu'à comparer le scheme aussi, port != ou scheme != pour moi ce n'est pas le même service.

Bien vu, merci ; voici une mise à jour avec une modif dans les tests (utiliser le "vrai" AnonymousUser et pas un mock)

Mais en fait je ne comprends pas trop le code de la classe Requests ça fait des trucs bizarre au lieu de simplement ajouter orig et la signature à l'URL ça va mélanger service[url] de dedans, je suppose qu'il y a une raison mais ce serait bien de la mettre dans le code à un jour.

Je suis assez d'accord que ce combo.utils.requests n'est pas bien lisible au premier coup d'oeil, il est perfectible ; mais pour l'instant je ne joue pas trop, mon objectif c'est de me faciliter la vie pour [#13122](#).

#8 - 09 septembre 2016 17:08 - Frédéric Péters

Sinon pour le reste ça me va. Mais en fait je ne comprends pas trop le code de la classe Requests ça fait des trucs bizarre au lieu de simplement ajouter orig et la signature à l'URL ça va mélanger service['url'] de dedans, je suppose qu'il y a une raison mais ce serait bien de la mettre dans le code à un jour.

Je veux bien essayer d'expliquer des trucs mais je ne vois pas à quoi tu fais référence. (je ne vois pas service['url'])

#9 - 09 septembre 2016 17:14 - Benjamin Dauvergne

```
remote_service_base_url = remote_service.get('url')
'''
scheme, netloc, old_path, params, old_query, fragment = urlparse.urlparse(
    remote_service_base_url)
...
path = url # je saute le cas où remote_url contient aussi une query string
...
url = urlparse.urlunparse((scheme, netloc, path, params, query, fragment))
```

Pourquoi n'utilise-t-on pas tout simplement url sans sa query string à laquelle on ajouterait orig et la signature ?

#10 - 09 septembre 2016 17:24 - Frédéric Péters

Parce qu'url peut en fait simplement être un chemin.

Dans les cellules wcs, le wcs concerné est sélectionné lors de la configuration de la cellule, et les appels aux webservices se font façon .get(wcs_site, '/api/...').

#11 - 09 septembre 2016 17:49 - Benjamin Dauvergne

Ok je comprends mieux.

#12 - 13 septembre 2016 15:49 - Thomas Noël

Ordoncques, le patch, il vous va ?

#13 - 13 septembre 2016 16:32 - Benjamin Dauvergne

Pour moi c'est ok.

#14 - 21 septembre 2016 13:20 - Thomas Noël

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

```
commit e9e2f34ed98a316781000b4466cf43bc0762e309
Author: Thomas NOEL <tnoel@entrouvert.com>
Date: Fri Sep 9 16:02:41 2016 +0200
```

```
utils.requests: remote_service can be guessed (#13125)
```

#15 - 23 décembre 2018 15:23 - Frédéric Péters

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

Fichiers

0001-utils.requests-remote_service-can-be-guessed-13125.patch	7,37 ko	09 septembre 2016	Thomas Noël
0001-utils.requests-remote_service-can-be-guessed-13125.patch	7,55 ko	09 septembre 2016	Thomas Noël