Passerelle - Development #16104

Avoir un transport SOAPpy qui loggue l'HTTP

28 avril 2017 16:06 - Frédéric Péters

Statut:	Fermé	**Début:**	28 avril 2017
Priorité:	Normal	**Echéance:**	
Assigné à:	Josué Kouka	**% réalisé:**	100%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

Après avoir nettoyé les soap.py dans le cadre de #13221, il sera pas mal que le code commun puisse logguer l'HTTP.

Demandes liées:

Lié à Passerelle - Bug #13221: multiplication des soap.py dans les connecteur...	**Nouveau**	**19 septembre 2016**
Lié à Passerelle - Development #24675: Utiliser python-zeep pour avoir du log...	**Fermé**	**20 juin 2018**
Lié à Passerelle - Development #24567: Connecteur IWS	**Fermé**	**15 juin 2018**

Révisions associées

Révision 2cc36d68 - 02 juillet 2018 10:21 - Emmanuel Cazenave

start a SOAP common infrastructure (#16104)

Historique

#1 - 28 avril 2017 16:06 - Frédéric Péters

- Lié à Bug #13221: multiplication des soap.py dans les connecteurs (nettoyage nécessaire) ajouté

#2 - 20 juin 2018 14:46 - Emmanuel Cazenave

- Assigné à mis à Emmanuel Cazenave

#3 - 20 juin 2018 17:18 - Emmanuel Cazenave

- Lié à Development #24675: Utiliser python-zeep pour avoir du log http ajouté

#4 - 25 juin 2018 12:36 - Frédéric Péters

Suite au commentaire 14 dans #24567 :

Je suis toujours pour une mise en commun, dans passerelle/utils/soap.py, que les connecteurs n'aient pas à faire cet import/configuration de Transport.

```
from passerelle.utils.api import Client
...
client = Client(connector=self) # pose le transport, tire wsdl_url de l'instance, etc.
```

#5 - 25 juin 2018 12:37 - Frédéric Péters

- Lié à Development #24567: Connecteur IWS ajouté

#6 - 25 juin 2018 12:43 - Emmanuel Cazenave

- Assigné à Emmanuel Cazenave supprimé

Je suis d'avis d'abandonner cette piste au profit de zeep.

#7 - 25 juin 2018 12:48 - Emmanuel Cazenave

J'avais pas vu le commentaire précédent.

#8 - 25 juin 2018 13:56 - Emmanuel Cazenave

Frédéric Péters a écrit :

> Je suis toujours pour une mise en commun, dans passerelle/utils/soap.py, que les connecteurs n'aient pas à faire cet import/configuration de Transport.

Si on parle bien ici d'un soap.py qui utiliserait zeep, je ne suis pas de ton avis.

Avec zeep, le log http s'obtient en deux lignes de code, c'est vraiment pas grand chose.

Ce soap.py devra à terme gérer tous les cas possibles d'initialisation d'un client SOAP, à comparer avec laisser les spécificités d'initialisation d'un client dans le connecteur qui l'utilise (ces spécificités étant dictés par le webservice qu'on a en face), je ne vois pas trop l'intérêt.

Mise à part le fait que ça aurait le mérite de standardiser les noms d'attributs de modèle de connecteur genre wsdl_endpoint...mais j'ai peur que la diversité des cas d'usage ne transforme rapidement ce soap.py en usine à gaz.

#9 - 25 juin 2018 14:07 - Frédéric Péters

Au contraire je trouve vraiment cettre normalisation utile et dans la logique de ce qui est déjà en place (attributs wsdl_url, verify_cert, client_certificate, etc.); qu'au moment où on se rend compte qu'un connecteur doit malheureusement faire de l'https sans vérifier le certificat, on sache qu'on peut juste ajouter un attribut verify_cert au modèle, sans devoir chercher dans la documentation ou copier/coller d'un autre connecteur le code touchant à zeep).

Qu'on puisse lire dans la documentation de zeep le conseil d'avoir un backend pour cacher des wsdl et xsd et qu'on ait à le faire qu'une fois, aussi.

#10 - 25 juin 2018 14:13 - Emmanuel Cazenave

- Assigné à mis à Emmanuel Cazenave

bon ok je le tente, même si je trouve ça dur à faire avec si peu de recul sur zeep.

#11 - 25 juin 2018 14:15 - Frédéric Péters

Merci; et comme je le notais, on peut/doit commencer tout léger, simplement tirer le wsdl_url et poser le transport, ce dont tu as besoin dans ton connecteur. Puis le travail de #13221 d'uniformisation ne disparaitra pas et au fur et à mesure de la conversion des connecteurs, passerelle/utils/soap.py pourra gagner en fonctionnalités.

#12 - 25 juin 2018 15:08 - Emmanuel Cazenave

Et du coup puisque l'on contraint ici l'attribut wsdl_url (pour commencer), je me dis que ça a intérêt à apparaître clairement dans modèle qui hérite de BaseResource , qui aurait une une property client etc...

#13 - 25 juin 2018 17:58 - Emmanuel Cazenave

- Fichier 0001-create-a-SOAP-common-infrastructure-16104.patch ajouté

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

Finalement opté pour le plus simple.

Je ne fais pas de soap_client une property, car le client SOAP peut alors être initialisé par le inspect.getmembers de passerelle/views.py::dispatch (et casser des tests sur des connecteurs dont le modèle définit un attribut wsdl_url, mais qui n'ont rien demandé).

#14 - 25 juin 2018 18:05 - Frédéric Péters

Ok pour moi, en mettant "start" plutôt que "create" dans le sujet du commit.

#15 - 25 juin 2018 18:08 - Emmanuel Cazenave

- Fichier 0001-create-a-SOAP-common-infrastructure-16104.patch ajouté

Ok, je pousserai ça.

#16 - 26 juin 2018 10:29 - Thomas Noël

J'arrive après la bataille, mais pourquoi mettre des trucs dans utils.py qui ne seront en fait jamais utilisés... Je pense qu'on pourrait tout poser dans le BaseResource, genre

```
    def get_soap_transport(self):
        return zeepTransport(session=self.requests, cache=zeepInMemoryCache())

    def get_soap_client(self):
        return zeepClient(self.wsdl_url, transport=self.get_soap_transport())
```

Nan ?

#17 - 28 juin 2018 15:37 - Josué Kouka

Vu que le patch n'est pas encore poussé. Peut on faire à ce que le l'initialisation du SOAPClient soit un peu plus flexible ?(Cas ou l'on a des Plugin ou Settings zeep à passer au client).

#18 - 28 juin 2018 16:06 - Emmanuel Cazenave

Avec plaisir mais tu ne m'en dis pas assez pour comprendre de dont tu as besoin, met un bout de ton code pour qu'on comprenne stp.

#19 - 28 juin 2018 16:12 - Josué Kouka

Emmanuel Cazenave a écrit :

> Avec plaisir mais tu ne m'en dis pas assez pour comprendre de dont tu as besoin, met un bout de ton code pour qu'on comprenne stp.

En fait je veux pouvoir juste faire:

```
settings = Settings(strict=False)
plugins = [FooPlugin, BarPluging]
client = self.soap_client(settings=settings, plugins=plugins)
client.service.foo(ham, eggs)
...
```

#20 - 29 juin 2018 09:23 - Josué Kouka

- Fichier 0001-start-a-SOAP-common-infrastructure-16104.patch ajouté

- Statut changé de En cours à Solution proposée

(Patch à la demande d'Emmanuel)

#21 - 02 juillet 2018 10:01 - Emmanuel Cazenave

- Statut changé de Solution proposée à Solution validée

- Assigné à changé de Emmanuel Cazenave à Josué Kouka

ok pour moi

#22 - 02 juillet 2018 10:22 - Josué Kouka

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

- % réalisé changé de 0 à 100

```
commit 2cc36d68aab4689ab4e85d1383457d2a0b0275e5 (HEAD -> master, origin/master, origin/HEAD)
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date:   Mon Jun 25 16:59:38 2018 +0200

    start a SOAP common infrastructure (#16104)
```

#23 - 04 août 2018 12:30 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-create-a-SOAP-common-infrastructure-16104.patch	8,26 ko	25 juin 2018	Emmanuel Cazenave
0001-create-a-SOAP-common-infrastructure-16104.patch	8,26 ko	25 juin 2018	Emmanuel Cazenave
0001-start-a-SOAP-common-infrastructure-16104.patch	8,64 ko	29 juin 2018	Josué Kouka