Passerelle - Development #65822

Création connecteur Signal Arrêtés

31 mai 2022 16:23 - Corentin Séchet

Statut:	Fermé	**Début:**	31 mai 2022
Priorité:	Normal	**Echéance:**	
Assigné à:	Corentin Séchet	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Logiciel de gestion des arrêtés de circulation

Révisions associées

Révision 2b14a8d6 - 04 juillet 2022 21:39 - Corentin Séchet

signal_arretes: create signal arretes connector (#65822)

Historique

#2 - 15 juin 2022 09:39 - Corentin Séchet

- Fichier 0001-signal_arretes-create-signal-arretes-connector-65822.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#3 - 15 juin 2022 12:00 - Thomas Noël

Tu peux remplacer if key in post_data and post_data[key]: par if post_data.get(key):

Tu devrais utiliser le mixin HTTPResource pour ta classe SignalArretes : tu gagneras les possibilités d'authentification sans effort.

Au niveau du « except RequestException » : il faut inclure les appels à self.request dans le try, car il peut y avoir un pépin de SSL, de timeout, etc.

(Manie perso, mais j'ai du mal avec les fonctions déclarées après leur utilisation :) C'est-à-dire que je trouve que _call, _get_list et _get_value pourraient être au début de la classe)

Sur les référentiels _get_list il faut être un peu plus avancer et parvenir à gérer les éventuels paramètres ?q=... et ?id=... : c'est nécessaire pour avoir l'autocomplétion qui marche dans les formulaires, c'est certainement utile pour les rues.

Sur les noms des endpoint, évitions les underscores, les "get", je propose de simplifier comme ça :

- get_cities -> cities
- get_lanes -> streets
- get_occupation_types -> occupations-types
- create_request -> create
- get_request_status -> status
- get_request_document -> document

Ca donnera des URL plus "propres" à utiliser dans w.c.s.

Tu noteras que je les liste dans l'ordre "naturel" d'usage, d'abord les référentiels, puis la création d'une demande, puis le suivi d'une demande.

En retour du create tu pourrais aussi renvoyer le statut reçu, on ne sait jamais, genre « request_id: D0000_DOT, request_status: Enregistré ».

Enfin, écrire toujours Signal Arrêtés avec S et A en majuscule. Et j'ajouterais bien un ™ sur « verbose_name = 'Signal Arrêtés™' » histoire de faire comprendre que c'est un logiciel spécifique (et pas un machin de d'arrêtés générique je ne sais quoi).

Voilà pour ma première passe !

#4 - 15 juin 2022 15:54 - Valentin Deniaud

Thomas Noël a écrit :

> Tu peux remplacer if key in post_data and post_data[key]: par if post_data.get(key):

Je dirais même plus, cette partie avec _update_data obscurcit pas mal ce qui est effectivement envoyé, je verrais bien tout à plat dans query_data, et si ensuite il y a à virer les valeurs vides, le faire en une ligne avec une boucle for {k: v for k, v in query_data if v}.

Peut-être lié, on ne fait nulle part du 'type': ['string', 'null'], et ça fait mauvais ménage dans la présentation de l'endpoint en BO : soit en revenir au 'type': 'string' classique, soit introduire un patch préliminaire pour gérer ça.

_format_date, que deux utilisations ==> pas de nécessité de factoriser, dupliquer ce code inline sera plus clair.

Je trouve les tests pas très limpides, ceci dit le point « on a 30 000 manières différentes de mocker des requêtes » a déjà été soulevé et il n'y a jamais eu de déter pour homogénéiser ou définir des bonnes pratiques, donc pas de nécessité à suivre ce que j'écris :

- Les mocks devraient être bêtes, juste servir une réponse à partir d'une URL, pas d'assert à l'intérieur.
  - Si il y a des choses à vérifier sur la requête émise elle est accessible à posteriori depuis l'objet mock en utilisant le décorateur @remember_called.
- Je trouve que les tests dans test_api_entreprise.py sont biens.
- Je trouve que test_create_request_errors est vraiment dur à lire, trop de niveaux d'indentation, trop de trucs qui s'appellent les uns les autres.

#5 - 18 juin 2022 01:59 - Corentin Séchet

- Fichier 0001-signal_arretes-create-signal-arretes-connector-65822.patch ajouté

> Thomas Noël a écrit :
> (Manie perso, mais j'ai du mal avec les fonctions déclarées après leur utilisation :) C'est-à-dire que je trouve que _call, _get_list et _get_value pourraient être au début de la classe)

Ma manie est de mettre les méthodes "privées" en fin de classe, c'est pour ça, mais va pour la tienne :)

> Valentin Deniaud a écrit
> Peut-être lié, on ne fait nulle part du 'type': ['string', 'null'], et ça fait mauvais ménage dans la présentation de l'endpoint en BO : soit en revenir au 'type': 'string' classique, soit introduire un patch préliminaire pour gérer ça.

Peut être qu'il y a une bonne manière de faire, mais je n'ai pas vu comment ne pas inclure un champ qui n'est pas défini côté WCS, lorsqu'on fait un appel WebService. Du coup si des champs ne sont pas définis, on a None, et mettre juste "string" provoquera une erreur côté passerelle puisque les données ne passeront pas la validation. S'il y a un moyen (simple) de ne pas envoyer les champs non définis, côté WCS, pourquoi pas, sinon je trouve ça bien de gérer le cas côté code. Tu parles d'un patch pour gérer la présentation en back office ?

> Les mocks devraient être bêtes, juste servir une réponse à partir d'une URL, pas d'assert à l'intérieur.

Je ne suis pas d'accord, je ne vois aucun intérêt à mettre les asserts sur les données envoyées par Passerelle en dehors des mocks. De mon côté, je trouve que si pour une raison ou une autre un mock est appelé (ajout d'un test, cuisine interne aux endpoints passerelle...), les données envoyées par passerelle seront testées et c'est bien. Ca évite de dupliquer le code qui vérifie les données à chaque appel. Si c'est juste pour renvoyer une réponse, utiliser une fonction n'a pas plus d'intérêt que d'utiliser mock.patch.

> Si il y a des choses à vérifier sur la requête émise elle est accessible à posteriori depuis l'objet mock en utilisant le décorateur @remember_called.

remember_called est très peu utilisé dans le code, non documenté (le deuxième résultat dans google à "remember_called httmock" c'est le code Entr'ouvert). Et je trouve ça très lourd comparé à juste tester la requête dans le mock, comme évoqué plus haut.

> Je trouve que les tests dans test_api_entreprise.py sont biens.

En effet, j'ai copié du coup :)

> Je trouve que test_create_request_errors est vraiment dur à lire, trop de niveaux d'indentation, trop de trucs qui s'appellent les uns les autres.

C'est relativement subjectif : il y a deux niveaux d'intentations, deux méthodes, aucun appel récursif : il y a pas mal de code qui risque d'être dur à lire à ce tarif. Si on parle en terme de complexité cyclomatique, utiliser une boucle for est même plus complexe que l'appel à une fonction imbriquée. Mais je comprends que ça soit un peu lourdingue, je ferais attention à ne pas abuser des fonctions imbriquées. J'ai aussi utilisé mock.patch, pour enlever un appel, c'est plus léger.

#6 - 20 juin 2022 15:06 - Valentin Deniaud

Corentin Séchet a écrit :

> je n'ai pas vu comment ne pas inclure un champ qui n'est pas défini côté WCS, lorsqu'on fait un appel WebService. Du coup si des champs ne sont pas définis, on a None, et mettre juste "string" provoquera une erreur côté passerelle puisque les données ne passeront pas la validation.

> S'il y a un moyen (simple) de ne pas envoyer les champs non définis, côté WCS, pourquoi pas

Je ne sais pas, interrogation à faire remonter à plus compétent que moi sur la question (Benjamin ?), ça m'a juste paru étrange que les autres connecteurs n'aient jamais eu à spécifier ça.

> Tu parles d'un patch pour gérer la présentation en back office ?

Ouep, l'affichage BO est un peu cassé j'ai l'impression.

> > Je trouve que les tests dans test_api_entreprise.py sont biens.
>
> En effet, j'ai copié du coup :)

Top !

> utiliser une boucle for est même plus complexe que l'appel à une fonction imbriquée.

Pour ce cas d'usage je pense qu'il faut utiliser pytest.mark.parametrize.

#7 - 24 juin 2022 10:36 - Corentin Séchet

- Fichier 0001-signal_arretes-create-signal-arretes-connector-65822.patch ajouté

- Statut changé de Solution proposée à Information nécessaire

- Assigné à changé de Corentin Séchet à Benjamin Dauvergne

J'ai utilisé pytest.mark.parametrize. Benjamin, si tu as un avis sur la question ci-dessus, ça nous intéresse.

#8 - 28 juin 2022 14:53 - Benjamin Dauvergne

- Assigné à changé de Benjamin Dauvergne à Corentin Séchet

#9 - 28 juin 2022 14:55 - Benjamin Dauvergne

- Assigné à changé de Corentin Séchet à Benjamin Dauvergne

#10 - 28 juin 2022 15:10 - Benjamin Dauvergne

- Statut changé de Information nécessaire à Solution validée

- Assigné à changé de Benjamin Dauvergne à Corentin Séchet

Valentin Deniaud a écrit :

> Corentin Séchet a écrit :
>
> > je n'ai pas vu comment ne pas inclure un champ qui n'est pas défini côté WCS, lorsqu'on fait un appel WebService. Du coup si des champs ne sont pas définis, on a None, et mettre juste "string" provoquera une erreur côté passerelle puisque les données ne passeront pas la validation. S'il y a un moyen (simple) de ne pas envoyer les champs non définis, côté WCS, pourquoi pas
>
> Je ne sais pas, interrogation à faire remonter à plus compétent que moi sur la question (Benjamin ?), ça m'a juste paru étrange que les autres connecteurs n'aient jamais eu à spécifier ça.

C'est parce que tu fais tes tests avec des expressions python coté w.c.s., avec des gabarits tu n'as jamais None, toujours une chaîne vide.

Pour le fond de la question, non il n'y a rien pour faire ça et effectivement comme la validation a lieu avant la consommation, on ne peut pas faire mieux que :

- se restreindre à n'utiliser que des gabarits coté w.c.s., ce qui est de toute façon la politique officiel de la secte EO&Co,
- ne pas déclarer le champ dans le schéma,
- mettre ['string', 'null'] qui est fait ailleurs en pagaille dans les connecteurs plus complexe contrairement à ce que dit Valentin, voir par exemple toulouse_axel. Mais en moins bien, j'avais l'habitude de faire des {'oneOf': [{'type': 'string'}, {'type': 'null'}]}, je ne connaissais pas cet usage de 'type', c'est effectivement plus court pour les types de base, oneOf devient nécessaire quand on a des validations plus complexes à combiner (genre object/null avec properties, pas sûr qu'on puisse utiliser type:object,null).

À noter qu'on peut utiliser aussi des références[1] pour raccourcir (mais je ne sais pas si c'est bien géré dans notre code de rendu des schéma, mais c'est un truc qu'on voit souvent en Swagger, donc prendre ma remarque avec des pincettes), ou bêtement une variable python

'STRING_OR_NULL_SCHEMA'[2].

[1] https://json-schema.org/understanding-json-schema/structuring.html

[2] https://git.entrouvert.org/passerelle.git/tree/passerelle/contrib/rsa13/models.py#n35

#11 - 28 juin 2022 15:15 - Valentin Deniaud

- Statut changé de Solution validée à En cours

Benjamin Dauvergne a écrit :

- se restreindre à n'utiliser que des gabarits coté w.c.s., ce qui est de toute façon la politique officiel de la secte EO&Co,
- mettre ['string', 'null']

Faire l'un ou l'autre, mais le second nécessite un patch préliminaire d'adaptation du rendu des schémas comme noté plus haut.

#12 - 28 juin 2022 15:24 - Benjamin Dauvergne

Valentin Deniaud a écrit :

> Benjamin Dauvergne a écrit :
>
> - se restreindre à n'utiliser que des gabarits coté w.c.s., ce qui est de toute façon la politique officiel de la secte EO&Co,
> - mettre ['string', 'null']
>
> Faire l'un ou l'autre, mais le second nécessite un patch préliminaire d'adaptation du rendu des schémas comme noté plus haut.

Ouaip, pas clair peut-être mais mon opinion c'est de mettre juste string et d'arrêter de faire des tests en live avec des expressions python.

#13 - 04 juillet 2022 21:40 - Corentin Séchet

- Fichier 0001-signal_arretes-create-signal-arretes-connector-65822.patch ajouté

- Statut changé de En cours à Solution proposée

#14 - 05 juillet 2022 13:59 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Go.

#15 - 05 juillet 2022 17:33 - Corentin Séchet

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

```
commit 2b14a8d6c61512bf742ab3f25c13d3472735e19f
Author: Corentin Séchet <csechet@entrouvert.com>
Date:   Wed Jun 1 15:12:51 2022 +0200

    signal_arretes: create signal arretes connector (#65822)
```

#16 - 08 juillet 2022 18:14 - Transition automatique

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

#17 - 11 septembre 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-signal_arretes-create-signal-arretes-connector-65822.patch	25,6 ko	15 juin 2022	Corentin Séchet
0001-signal_arretes-create-signal-arretes-connector-65822.patch	28,1 ko	17 juin 2022	Corentin Séchet
0001-signal_arretes-create-signal-arretes-connector-65822.patch	28,1 ko	24 juin 2022	Corentin Séchet
0001-signal_arretes-create-signal-arretes-connector-65822.patch	28 ko	04 juillet 2022	Corentin Séchet