

Passerelle - Development #33099

developper le connecteur Kimoce

15 mai 2019 07:11 - Serghei Mihai

Statut:	Fermé	Début:	15 mai 2019
Priorité:	Normal	Echéance:	
Assigné à:	Serghei Mihai	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Connecteur spécifique pour s'interfacer avec l'API développée en interne par la ville de Lille.			

Révisions associées

Révision 1cb079e4 - 21 mai 2019 18:04 - Serghei Mihai

lille_kimoce: initial connector (#33099)

Historique

#2 - 15 mai 2019 07:17 - Frédéric Péters

Dont l'objectif est ?

#3 - 15 mai 2019 07:35 - Serghei Mihai

Le but est de remonter les demandes de signalement de propreté dans Kimoce.

L'API expose des référentiels des catégories, types et sous-types de signalement, ainsi que la liste des voies qui pourrait être utilisée dans le formulaire pour l'aide à la saisie de l'adresse.

Avec ces informations on crée une demande dans Kimoce.

L'authentification auprès de l'API se fait par login/mot de passe qui délivre un jeton utilisé en authentification "Bearer":

<https://github.com/lexik/LexikJWTAuthenticationBundle/blob/master/Resources/doc/index.md#usage>

Après le traitement et la cloture de la demande Kimoce viendra notifier wcs de l'avancement via un appel à trigger.

#4 - 15 mai 2019 16:21 - Serghei Mihai

- Fichier 0001-lille_kimoce-initial-connector-33099.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#5 - 15 mai 2019 16:25 - Serghei Mihai

Code écrit pendant le 2 jours de sprint, testé avec l'API développée par dessus de Kimoce.

#6 - 20 mai 2019 11:36 - Thomas Noël

Pose self.check_status plutôt au début du modèle qu'au milieu des endpoint normaux (genre juste après get_token). Mais surtout j'avoue ne pas comprendre comment il va marcher : le get sur les catégories nécessite un token. À mon avis tu devrais limiter le check_status à l'obtention du token.

J'ai cru que self.request permettait de faire n'importe quelle requête, mais non, c'est juste pour attraper certaines listes et donc je l'appellerai plutôt self.get_list ou self.referential

Détail, mais ça serait bien de documenter l'affaire : picture1 et picture2, quels content_type sont possibles ? Aussi, on aura à utiliser cela dans wcs (on n'y a pas de possibilité d'envoyer du base64 uniquement), le serialiseur de fichiers envoie plutôt un dictionnaire avec filename/content_type/base64.

Pour les listes, tu pourrais implémenter ?q et ?id afin de rendre l'autocomplétion directement fonctionnelle.

Sur la création d'une demande, tu devrais renvoyer aussi le contenu de la réponse (en retirant ou réduisant quand même pictures parce que ça va bouffer de la place). Ça permettra de simplifier le support, de voir ce que le Kimoce a répondu.

Dans les tests, tu pourrais réduire la taille des exemples :) Mais aussi supprimer des trucs comme "adrien.millot@smile.fr" parce que bon.

Voilà voilà.

#7 - 20 mai 2019 11:37 - Thomas Noël

Aussi, comme a dit Frédéric, utiliser le six de Django (si le urljoin de Django ne fait pas l'affaire, d'ailleurs ?). Et virer le import six qui ne sert à rien dans les tests.

#8 - 20 mai 2019 19:36 - Serghei Mihai

- Fichier 0001-lille_kimoce-initial-connector-33099.patch ajouté

Thomas Noël a écrit :

Pose self.check_status plutôt au début du modèle qu'au milieu des endpoint normaux (genre juste après get_token). Mais surtout j'avoue ne pas comprendre comment il va marcher : le get sur les catégories nécessite un token. À mon avis tu devrais limiter le check_status à l'obtention du token.

J'ai cru que self.request permettait de faire n'importe quelle requête, mais non, c'est juste pour attraper certaines listes et donc je l'appellerais plutôt self.get_list ou self.referential

Yep.

Détail, mais ça serait bien de documenter l'affaire : picture1 et picture2, quels content_type sont possibles ? Aussi, on aura à utiliser cela dans wcs (on n'y a pas de possibilité d'envoyer du base64 uniquement), le serialiseur de fichiers envoie plutôt un dictionnaire avec filename/content_type/base64.

L'api en face attend des photos, encodées en base64, sans content-type, sans nom car elle convertit le contenu en JPG et nomme les fichiers de son côté, d'après ses règles.

Pour les listes, tu pourrais implémenter ?q et ?id afin de rendre l'autocomplétion directement fonctionnelle.

Oui, fait.

Sur la création d'une demande, tu devrais renvoyer aussi le contenu de la réponse (en retirant ou réduisant quand même pictures parce que ça va bouffer de la place). Ça permettra de simplifier le support, de voir ce que le Kimoce a répondu.

Ok, mais en fait en retour le WS renvoie les noms des images créées, qui ne nous sert pas, et non leur contenu.

#9 - 21 mai 2019 10:05 - Thomas Noël

Serghei Mihai a écrit :

L'api en face attend des photos, encodées en base64, sans content-type, sans nom car elle convertit le contenu en JPG et nomme les fichiers de son côté, d'après ses règles.

Mais en fait je me demande comment tu vas envoyer du base64 là dedans...?

Sur la création d'une demande, tu devrais renvoyer aussi le contenu de la réponse (en retirant ou réduisant quand même pictures parce que ça va bouffer de la place). Ça permettra de simplifier le support, de voir ce que le Kimoce a répondu.

Ok, mais en fait en retour le WS renvoie les noms des images créées, qui ne nous sert pas, et non leur contenu.

Impec.

J'avais raté le passerelle/utils/http_authenticators.py : en faire un premier patch séparé stp.

Aussi, un truc, le response.ok c'est assez large comme vérification, ça répond aussi ok pour un 302 par exemple. Faudrait pas plutôt tester que c'est un 2xx voire même 200 ?

#10 - 21 mai 2019 10:33 - Serghei Mihai

- Fichier 0002-lille_kimoce-initial-connector-33099.patch ajouté

- Fichier 0001-utils-add-simple-HTTP-Bearer-authentication-class.patch ajouté

Thomas Noël a écrit :

Mais en fait je me demande comment tu vas envoyer du base64 là dedans...?

Dans l'appel WS par wcs on fait comme d'habitude: form_var_fichier_raw qui sera sérialisé en dico avec filename, content et content-type. Et le connecteur ne prend que le content.

J'avais raté le passerelle/utils/http_authenticators.py : en faire un premier patch séparé stp.

Ok.

Aussi, un truc, le response.ok c'est assez large comme vérification, ça répond aussi ok pour un 302 par exemple. Faudrait pas plutôt tester que c'est un 2xx voire même 200 ?

Lors de la création d'une demande l'API en face renvoie une HTTP 201. Pas de 302 rencontrée lors des tests.

#11 - 21 mai 2019 10:52 - Thomas Noël

Serghei Mihai a écrit :

Thomas Noël a écrit :

Mais en fait je me demande comment tu vas envoyer du base64 là dedans...?

Dans l'appel WS par wcs on fait comme d'habitude: form_var_fichier_raw qui sera sérialisé en dico avec filename, content et content-type. Et le connecteur ne prend que le content.

Rah, my bad, je viens de mieux relire et voir le « post_data[param_name]['content'] ». Je pensais qu'on pouvait décrire le truc dans le schéma JSON ?

Si on peut pas, il faut renforcer un peu le check, parce que ça va facilement plantouiller :

```
if post_data.get(param_name) and isinstance(post_data[param_name], dict) and post_data[param_name].get('content'):
    payload['pictures'].append({'content': post_data[param_name]['content']})
```

Aussi, un truc, le response.ok c'est assez large comme vérification, ça répond aussi ok pour un 302 par exemple. Faudrait pas plutôt tester que c'est un 2xx voire même 200 ?

Lors de la création d'une demande l'API en face renvoie une HTTP 201. Pas de 302 rencontrée lors des tests.

Donc c'est bien status_code // 100 == 2 qu'il faut tester (ie les 3xx sont un échec). Mais bon c'est un détail.

#12 - 21 mai 2019 11:48 - Serghei Mihai

- Fichier 0001-lille_kimoce-initial-connector-33099.patch ajouté

Thomas Noël a écrit :

Rah, my bad, je viens de mieux relire et voir le « post_data[param_name]['content'] ». Je pensais qu'on pouvait décrire le truc dans le schéma JSON ?

JSON schéma permet de définir un type "object", mais si les photos, facultatives, ne sont pas chargées, dans l'appel WS c'est null qui est transmis dans le payload json.

J'applique ta suggestion.

Donc c'est bien status_code // 100 == 2 qu'il faut tester (ie les 3xx sont un échec). Mais bon c'est un détail.

Ok, fait.

#13 - 21 mai 2019 16:00 - Thomas Noël

Désolé de relire mal, encore un truc que je trouve un peu dommage : on devrait remonter tout ce qui vient de leur référentiels et juste ajouter id/text. Par exemple sur les catégories, on reçoit :

```
{ "@id": "/api/categories/70",
  "id": 70,
  "label": "Table surface",
  "reference": "TBS" }
```

```
},
```

On va prendre ici l'info "reference" qui ne sera pas remontée. Et si un jour Kimoce ajoute des choses, on ne les remontera pas non plus.

Selon moi, on devrait remonter un truc avec juste l'ajout de "text" :

```
{
  "@id": "/api/categories/70",
  "id": 70,
  "label": "Table surface",
  "text": "Table surface",
  "reference": "TBS"
},
```

C'est-à-dire, pour chaque entrée d'un référentiel :

- s'il y a déjà un id, ne pas y toucher ; sinon l'ajouter avec la valeur de @id
- s'il y a déjà un text, ne pas y toucher, sinon l'ajouter avec la valeur de "label" ou de "id" (du premier qui existe)

Je ferais ça sur get_referential et sur streets, surtout dans l'idée que tout cela pourrait grandir chez Kimoce (et le connecteur grandira avec, tranquille)

A relire streets, je comprends pas bien le passage :

```
if 'id' in kwargs:
    params['streetAddress'] = kwargs['id']
if 'q' in kwargs:
    params['streetAddress'] = kwargs['q']
```

et je pense qu'il y a un bogue au niveau de la ligne 193, la variable "q" n'existe pas (il faut refaire la même requête qu'au dessus, juste renouveler le token)

#14 - 21 mai 2019 17:00 - Serghei Mihai

- Fichier 0001-lille_kimoce-initial-connector-33099.patch ajouté

Thomas Noël a écrit :

Selon moi, on devrait remonter un truc avec juste l'ajout de "text" :

[...]

C'est-à-dire, pour chaque entrée d'un référentiel :

- s'il y a déjà un id, ne pas y toucher ; sinon l'ajouter avec la valeur de @id
- s'il y a déjà un text, ne pas y toucher, sinon l'ajouter avec la valeur de "label" ou de "id" (du premier qui existe)

Ok, je rajoute ça à l'exception de l'id. La vraie valeur qui sera utilisé à la création d'une demande dans Kimoce (le _raw) est la valeur de @id. Je préfère éviter que les fonctionnels tapent dans l'appel WS {{ form_var_type_@id }}. L'attribut id sert à filtrer les types, sous-types en fonction du parent. (no comment)

A relire streets, je comprends pas bien le passage :

[...]

Bien vu, corrigé.

#15 - 21 mai 2019 17:58 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Pour moi on prend un peu de risque à faire des choses genre member['text'] = member['label'] (parce qu'un jour un référentiel n'aura pas de label), d'une façon générale ce connecteur est assez confiant sur les formats des retours :) Mais on peut aussi avoir confiance, à un moment, et donc ça me va d'en rester là pour l'instant.

J'aurais bien vu une factorisation des répétitifs « requests(token), si 401: requests(new_token) » (surtout que si y'a un 401 ensuite, on fait un peu n'importe quoi) ; mais ça peut attendre aussi.

Ack ainsi en version à poser en recette.

#16 - 21 mai 2019 18:04 - Serghei Mihai

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

commit 1cb079e4ba888d3624f0463cad3a825c37f18c65 (HEAD -> master, origin/master, origin/HEAD)

Author: Serghei Mihai <smihai@entrouvert.com>

Date: Tue May 14 09:47:44 2019 +0200

lille_kimoce: initial connector (#33099)

commit a7e78140021c170b30435dd37ab1926f23323db8

Author: Serghei Mihai <smihai@entrouvert.com>

Date: Tue May 21 10:26:32 2019 +0200

utils: add simple HTTP Bearer authentication class

#17 - 21 mai 2019 20:15 - Frédéric Péters

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

Fichiers

0001-lille_kimoce-initial-connector-33099.patch	30,5 ko	15 mai 2019	Serghei Mihai
0001-lille_kimoce-initial-connector-33099.patch	25 ko	20 mai 2019	Serghei Mihai
0001-utils-add-simple-HTTP-Bearer-authentication-class.patch	1,06 ko	21 mai 2019	Serghei Mihai
0002-lille_kimoce-initial-connector-33099.patch	24,2 ko	21 mai 2019	Serghei Mihai
0001-lille_kimoce-initial-connector-33099.patch	24,5 ko	21 mai 2019	Serghei Mihai
0001-lille_kimoce-initial-connector-33099.patch	24,9 ko	21 mai 2019	Serghei Mihai