

Passerelle - Development #27653

Connecteur planitech

29 octobre 2018 18:08 - Emmanuel Cazenave

Statut:	Fermé	Début:	29 octobre 2018
Priorité:	Normal	Echéance:	
Assigné à:	Emmanuel Cazenave	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Un connecteur pour planitech : #23958			

Révisions associées

Révision 073d35bd - 19 novembre 2018 16:28 - Emmanuel Cazenave

create planitech connector (#27653)

Historique

#1 - 29 octobre 2018 18:13 - Emmanuel Cazenave

- Statut changé de Nouveau à En cours

- Assigné à mis à Emmanuel Cazenave

#2 - 07 novembre 2018 19:44 - Emmanuel Cazenave

Pas encore prêt mais ça commence à ressembler à quelque chose.

Si quelqu'un peut déjà jeter un œil ce serait super (parce déjà 700 lignes ... joie).

#3 - 07 novembre 2018 20:35 - Frédéric Péters

Lecture rapide,

```
# passerelle - uniform access to multiple data sources and services
# Copyright (C) 2018 Entr'ouvert
```

En début de fichier. (et un jour faudra corriger la quantité de fichiers qui ne suivent pas ça).

```
+TZ = timezone('Europe/Paris') # FIXME, configuration ?
```

Au cas où le debian/settings.py pose déjà TIME_ZONE = 'Europe/Paris'.

```
+ raise APIError("Invalid time format : %s" % time_str, log_error=True)
```

Pas d'espace avant : en anglais.

Tu fais systématiquement log_error=True. Est-ce parce que tu as une utilisation particulière de cette exception, ou parce que tu penses que ça devrait être le cas partout ? (je serais alors pour discuter de l'adaptation du comportement par défaut).

```
+def _parse_time(time_str):
...
+def build_createreservation_params(data):
+ date_str = data['date']
+ start_date = parse_date(date_str)
+ if start_date is None:
+ raise APIError("Invalid date string: %s" % date_str, log_error=True)
```

J'adopterais la même pratique pour parse_date et parse_time, plutôt que factoriser parse_time mais répéter la gestion d'erreur du parse_date.

```
+ keep_cookies = models.BooleanField(
+ verbose_name=_('Keep HTTP cookies'), default=True)
```

Il y a des raisons pour parfois être mis à False ?

```
+ verify_cert = models.BooleanField(
```

```
+         default=True, verbose_name=_('Check HTTPS Certificate validity'))
```

Si toutes les instances rencontrées avaient un certificat valide, je zapperais ce paramètre.

```
+     class Meta:
+         verbose_name = _('Planitech connector')
```

Ne répétons pas "connector" dans les noms des connecteurs. (oui ici aussi il y aurait quelques connecteurs à corriger)

```
+     def _login(self):
+     ...
+         response = self.requests.get(auth_url, headers={'MH-PASSWORD': hash_pass})
```

Je trouve curieux de prendre la response puis que la fonction se termine sans rien en faire.

```
+     def createreservation(self, request):
+         data = validate_json(request, CREATE_RESERVATION_SCHEMA)
```

C'est voyant ça que j'ai attaché un patch rapide à [#25590](#). Si on décide d'adopter json schema, faisons-le officiellement, encourageons la pratique, partageons le code.

```
+     def daily(self):
```

Ça m'ennuie d'avoir ce code là-dedans; je préférerais une méthode au nom explicite, genre `cache_places_info`, et le `daily` qui l'appelle, si ça a un sens.

```
+     @endpoint(methods=['get'], perm='can_access')
+     def updateplaces(self, request):
+         self.daily()
```

Ou `updateplaces` comme nom explicite, même si j'aime moins.

À noter aussi que `w.c.s.` dispose (désormais) d'un cache sur les datasources, peut-être que ça rend inutile de l'avoir côté Passerelle.

```
+++ b/passerelle/contrib/planitech/mste.py
@@ -0,0 +1,267 @@
```

no comment sur ce contenu, mais quand même tapé notre entête passerelle/licence en haut de fichier.

```
+     def getfreegaps(self, request):
```

Je ne sais pas quel va être l'usage du connecteur, ni précisément ce que retourne cet endpoint, mais si jamais c'est proche de l'idée "créneau libre" de `chrono`, pour profiter automatiquement du widget adapté côté `w.c.s.`, le contenu pourrait être transformé pour être équivalent à ce que produit `chrono`.

#4 - 08 novembre 2018 15:56 - Thomas Noël

(revu avec Emmanuel, simplification drastique de l'affaire à faire, un nouveau plan est en cours qui devrait rendre la lecture du code plus claire)

#5 - 12 novembre 2018 16:52 - Frédéric Péters

1. passerelle - uniform access to multiple data sources and services

(attention au `s` final, absent de la branche)

#6 - 12 novembre 2018 17:06 - Emmanuel Cazenave

- Fichier `0001-create-planitech-connector-27653.patch` ajouté
- Statut changé de *En cours* à *Solution proposée*
- Patch *proposed* changé de *Non* à *Oui*

Frédéric Péters a écrit :

Tu fais systématiquement `log_error=True`.

Copier/coller bête sans réfléchir à ce que je faisais, j'ai tout enlevé.

Il y a des raisons pour parfois être mis à `False` ?

Reliquat de ma première tentative dans [#27654](#), supprimé.

Si toutes les instances rencontrées avaient un certificat valide, je zapperais ce paramètre.

La seule instance à laquelle j'ai accès pour l'instant n'a pas de certificat valide.

Je trouve curieux de prendre la response puis que la fonction se termine sans rien en faire.

J'ai mis un commentaire dans le code : le but du jeu est d'obtenir un cookie. Mais j'ai constaté empiriquement ([#27720](#)) que l'obtention d'un cookie ne garantit rien (il peut-être posé sans que l'authentification ait bien fonctionné), le juge de paix est l'utilisation de l'API métier proprement dite qui renvoie bien des 403 si l'authentification n'a pas fonctionné (et cela est géré dans `_call_planitech`).

Commentaire connexe : les appels à `_login` sont maintenant cachés dans `_call_planitech` pour qu'on ait pas à s'en soucier dans tous les endpoint.

C'est voyant ça que j'ai attaché un patch rapide à [#25590](#). Si on décide d'adopter json schema, faisons-le officiellement, encourageons la pratique, partageons le code.

Rebasé mon patch sur le tien.

(attention au s final, absent de la branche)

Corrigé.

Je ne sais pas quel va être l'usage du connecteur, ni précisément ce que retourne cet endpoint, mais si jamais c'est proche de l'idée "créneau libre" de chrono, pour profiter automatiquement du widget adapté côté w.c.s., le contenu pourrait être transformé pour être équivalent à ce que produit chrono.

Sur les conseils avisés de Thomas (parce que j'étais en train de me perdre), et sur l'idée que ce qui intéresse les gens c'est de réserver une salle pour la date X : deux endpoints utilisables comme des data source, `getdays` et `getplaces`.

`getdays` pour connaître les dates possibles de réservation (outre les critères temporels, on peut ajouter des critères de capacité de salle).
`getplaces` qui renvoie les salles disponibles sur une date précise (qui prend aussi les restrictions sur les capacités de salle).

Je pense avoir tenu compte des toutes tes autres remarques. Et ajout de tests fonctionnels.

#7 - 13 novembre 2018 11:03 - Emmanuel Cazenave

- Fichier `0001-create-planitech-connector-27653.patch` ajouté

Dans éclair de lucidité, utilisation plus orthodoxe du cache.

```
--- a/passerelle/contrib/planitech/models.py
+++ b/passerelle/contrib/planitech/models.py
@@ -217,7 +217,7 @@ class PlanitechConnector(BaseResource):
     )
     for place in data['requestedPlaces']:
         ref[place['identifiant']]['capacity'] = place.get('capacity')
-    cache.set(cache_key, ref, None)
+    cache.set(cache_key, ref)
```

#8 - 13 novembre 2018 14:54 - Emmanuel Cazenave

- Fichier `0001-create-planitech-connector-27653.patch` ajouté

Mini ajustement sur le `parse_date` et `parse_time`, avec les tests qui vont avec.

#9 - 14 novembre 2018 08:36 - Frédéric Péters

Je marquerai pour traduction les descriptions des endpoints.

Quid des FIXME qui restent dans le code ?

#10 - 14 novembre 2018 12:24 - Emmanuel Cazenave

- Fichier `0001-create-planitech-connector-27653.patch` ajouté

Frédéric Péters a écrit :

Je marquerai pour traduction les descriptions des endpoints.

Voilà.

Quid des FIXME qui restent dans le code ?

Je suis sur l'idée de livrer ça et d'affiner ensuite après une recette avec Vénissieux : je ne sais pas de quelle niveau de finesse on aura besoin pour les points marqués en FIXME (genre un champ du modèle fera l'affaire ou cela devra être géré par un paramètre du endpoint ?).

#11 - 19 novembre 2018 14:27 - Frédéric Péters

- Statut changé de *Solution proposée* à *Solution validée*

Ok mais il faut refaire la migration initiale pour suivre [#27162](#). (et j'attendrais un build réussi avec [#27162](#) avant de pousser)

#12 - 19 novembre 2018 16:40 - Emmanuel Cazenave

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

```
commit 073d35bd8ab3b30c7dc60a29442de3a49541fffc
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date: Fri Nov 9 16:36:24 2018 +0100
```

```
create planitech connector (#27653)
```

#13 - 30 novembre 2018 12:17 - Emmanuel Cazenave

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

#14 - 03 décembre 2018 15:15 - Benjamin Dauvergne

- Statut changé de *Solution déployée* à *Fermé*

Fichiers

web-services-api.pdf	570 ko	29 octobre 2018	Emmanuel Cazenave
exemple-php.zip	20,2 ko	29 octobre 2018	Emmanuel Cazenave
0001-create-planitech-connector-27653.patch	48,4 ko	12 novembre 2018	Emmanuel Cazenave
0001-create-planitech-connector-27653.patch	48,4 ko	13 novembre 2018	Emmanuel Cazenave
0001-create-planitech-connector-27653.patch	49,6 ko	13 novembre 2018	Emmanuel Cazenave
0001-create-planitech-connector-27653.patch	49,6 ko	14 novembre 2018	Emmanuel Cazenave