Passerelle - Development #29514

planitech : gestion des utilisateurs

07 janvier 2019 12:35 - Emmanuel Cazenave

Statut:	Fermé	**Début:**	07 janvier 2019
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

De #29136 : "le réservant (non pris en compte)". Et pour cause on passe pour l'instant un contractorExternalIdentifier fantaisiste.

Après un peu d'exploration de l'API et de la GUI Planitech voilà mon plan.

- restriction des démarches utilisant ce connecteur à des usagers connectés
- nouveau endpoint **getorcreateuser** qui passe en contractorExternalIdentifier l'uuid de l'utilisateur connecté (passé en paramètre de la requête), plus nom prénom mail.
  - en interne ce endpoint tape dans un cache utilisateurs pour vérifier l'existence de l'utilisateur
- tout appel a **createcreservation** doit être précédé d'un appel **getorcreateuser** pour assurer l'existence de l'utilisateur dans planitech
- l'appel à **createcreservation** gagne un paramètre uuid qu'il positionne sur le contractorExternalIdentifier

Ça permettrait d'avoir un affichage bien propre dans Planitech, avec un utilisateur associé à chaque réservation, de croiser les infos Planitech Publik, et c'est pas bien compliqué.
Dans Planitech il y a de l'isolation au niveaux des utilisateurs : l'utilisateur technique (qui a les droits de connexion à l'API) ne voit que les utilisateurs métiers qu'il a créé lui même.

Le plan alternatif était d'avoir un utilisateur unique auquel Publik rattache toutes les réservations qu'il fait, moche, pas de recoupement avec Publik, etc ...

Révisions associées

Révision bb8fee66 - 15 janvier 2019 09:40 - Emmanuel Cazenave

planitech: create users under the hood (#29514)

Historique

#1 - 07 janvier 2019 14:44 - Benjamin Dauvergne

Idéalement on ne devrait pas passer l'identifiant Publik à Planitech mais utiliser un pseudonyme, soit un hash de l'uuid, soit générér un nouvel uuid et conserver dans un modèle le lien uuid Publik -> uuid Planitech.

#2 - 07 janvier 2019 14:47 - Benjamin Dauvergne

Je simplifierai l'API ainsi :

- ne pas ajouter de nouveau endpoint
- avoir un modèle pour se souvenir de la création des utilisateurs
- passer systématiquement à createreservation nom, prénom, email, uuid; à charge pour ce endpoint de voir si l'utilisateur existe ou pas de l'autre coté (le modèle du point précédent devrait faciliter les choses) et d'en faire la création si besoin

#3 - 08 janvier 2019 11:00 - Emmanuel Cazenave

Benjamin Dauvergne a écrit :

> Idéalement on ne devrait pas passer l'identifiant Publik à Planitech mais utiliser un pseudonyme, soit un hash de l'uuid, soit générér un nouvel uuid et conserver dans un modèle le lien uuid Publik -> uuid Planitech.

Pour des considérations de sećurité j'imagine ?

Pour la simplification du endpoint j'y avais pensé et j'hésite, si un autre client nous demande des créer toutes les réservations avec un utilisateur unique ou autre, il faudrait refaire ...

#4 - 09 janvier 2019 17:27 - Emmanuel Cazenave

- Fichier 0001-planitech-create-users-under-the-hood-29514.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

J'ai tout fait comme t'as dit Benj, c'est quand même bien plus commode d'utilisation.

#5 - 09 janvier 2019 18:18 - Benjamin Dauvergne

C'est mon coté casse couille, mais si l'objectif c'est une unicité (resource, name_id) alors il vaudrait mieux faire comme cela:

- ajouter un index d'unicité sur (resource, name_id)
- réécrire ça:

```
+        try:
+            pairing = Pairing.objects.get(resource=self, name_id=post_data['name_id'])
+        except Pairing.DoesNotExist:
+            pairing = Pairing.objects.create(
+                resource=self,
+                name_id=post_data['name_id'],
+                external_id=uuid.uuid4().get_hex()
+            )
+            params = {
+                "externalUserIdentifier": pairing.external_id,
+                "name": post_data['last_name'],
+                "firstName": post_data['first_name'],
+                "mail": post_data['email']
+            }
+            try:
+                data = self._call_planitech(self.requests.post, 'createPerson', params)
+                if data.get('creationStatus') != 'OK':
+                    raise APIError("Person creation failed: %s" % data.get('creationStatus'))
+            except APIError:
+                pairing.delete()
+                raise
```

en ça

```
+        with atomic():
+            pairing, created = Pairing.objects.get_or_create(resource=self, name_id=post_data['name_id'],
+                defaults={'external_id': uuid.uuid4().get_hex()})
+            if created:
+                params = {
+                    "externalUserIdentifier": pairing.external_id,
+                    "name": post_data['last_name'],
+                    "firstName": post_data['first_name'],
+                    "mail": post_data['email']
+                }
+                data = self._call_planitech(self.requests.post, 'createPerson', params)
+                if data.get('creationStatus') != 'OK':
+                    raise APIError("Person creation failed: %s" % data.get('creationStatus'))
```

On est sûr que des appels concurrents ne vont pas foirer, et le with atomic() devrait garantir qu'on ait un lock implicite sur la ligne, en gros soit on crée le truc et si l'appel à l'API marche ça reste, soit on récupère une ligne existante et ça implique que l'appel l'API a fonctionné (grâce à atomic).

#6 - 10 janvier 2019 12:27 - Emmanuel Cazenave

- Fichier 0001-planitech-create-users-under-the-hood-29514.patch ajouté

Yes beaucoup plus propre, merci (pourtant je connais atomic, où avais-je la tête).

Et tant qu'à faire une deuxième contrainte unique_together sur ('resource', 'external_id').

#7 - 10 janvier 2019 12:34 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Emmanuel Cazenave a écrit :

Et tant qu'à faire une deuxième contrainte unique_together sur ('resource', 'external_id').

Yep bonne idée, les contraintes en base ça me fait plaisir.

Ack.

#8 - 15 janvier 2019 09:43 - Emmanuel Cazenave

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

```
commit bb8fee662b54718e72104c1ff3cea854ae9a2532 (HEAD -> master, origin/master, origin/HEAD)
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date:   Wed Jan 9 17:25:01 2019 +0100

    planitech: create users under the hood (#29514)
```

#9 - 17 janvier 2019 14:16 - Frédéric Péters

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

#10 - 12 avril 2019 10:27 - Benjamin Dauvergne

- Statut changé de Solution déployée à Fermé

Fichiers

0001-planitech-create-users-under-the-hood-29514.patch	10,2 ko	09 janvier 2019	Emmanuel Cazenave
0001-planitech-create-users-under-the-hood-29514.patch	10,6 ko	10 janvier 2019	Emmanuel Cazenave