

Passerelle - Development #69886

connecteur smsfactor fail cron update_credit_left (KeyError: 'credits')

04 octobre 2022 20:06 - Sentry Io

Statut:	Fermé	Début:	04 octobre 2022
Priorité:	Normal	Echéance:	
Assigné à:	A. Berriot	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
https://sentry.entrouvert.org/entrouvert/gplexpert/issues/95762/			
<pre>KeyError: 'credits' File "passerelle/base/management/commands/cron.py", line 59, in handle getattr(connector, frequency)() File "passerelle/apps/smsfactor/models.py", line 203, in hourly self.update_credit_left() File "passerelle/apps/smsfactor/models.py", line 173, in update_credit_left self.credit_left = result['credits'] connector "smsfactor.sms-via-sms-factor" error running hourly job</pre>			

Révisions associées

Révision 78df755c - 06 octobre 2022 14:59 - A. Berriot

smsfactor: fixed crash when updating credit (#69886)

Historique

#1 - 04 octobre 2022 20:07 - Lauréline Guérin

- *Projet changé de Suivi des traces à Passerelle*
- *Sujet changé de KeyError: 'credits' à connecteur smsfactor fail cron update_credit_left (KeyError: 'credits')*

#2 - 05 octobre 2022 09:11 - A. Berriot

- *Assigné à mis à A. Berriot*

#3 - 05 octobre 2022 09:16 - A. Berriot

- *Fichier 0001-smsfactor-fixed-crash-when-updating-credit-69886.patch ajouté*
- *Tracker changé de Bug à Development*
- *Statut changé de Nouveau à Solution proposée*
- *Patch proposed changé de Non à Oui*

#4 - 05 octobre 2022 09:16 - A. Berriot

- *Assigné à changé de A. Berriot à Benjamin Dauvergne*

(le problème de fond ici est que le token est pas bon mais mon patch gère ce cas pour éviter un crash côté sentry)

#5 - 05 octobre 2022 09:49 - Valentin Deniaud

- *Statut changé de Solution proposée à Solution validée*
- *Assigné à changé de Benjamin Dauvergne à A. Berriot*

#6 - 05 octobre 2022 09:53 - Thomas Noël

Agate Berriot a écrit :

(le problème de fond ici est que le token est pas bon)

Juste pour parler : ce n'est pas possible de détecter cela au niveau de self.request ?

#7 - 05 octobre 2022 10:02 - A. Berriot

- Fichier 0001-smsfactor-fixed-crash-when-updating-credit-69886.patch ajouté

- Statut changé de Solution validée à Solution proposée

#8 - 05 octobre 2022 10:02 - A. Berriot

Thomas Noël a écrit :

Agate Berriot a écrit :

(le problème de fond ici est que le token est pas bon

Juste pour parler : ce n'est pas possible de détecter cela au niveau de self.request ?

Si carrément, c'est même beaucoup mieux :D

mon second patch fait ça ;)

#9 - 05 octobre 2022 10:22 - A. Berriot

- Fichier 0001-smsfactor-fixed-crash-when-updating-credit-69886.patch ajouté

#10 - 05 octobre 2022 10:49 - A. Berriot

- Fichier 0001-smsfactor-fixed-crash-when-updating-credit-69886.patch ajouté

#11 - 05 octobre 2022 10:50 - A. Berriot

Bon en fait y'avait des erreurs dans les tests un peu moins triviales à gérer (notamment le multi destinataire). Avec un peu de chance ça passe sur jenkins et tout le monde est content.

#12 - 05 octobre 2022 12:32 - Thomas Noël

```
raise_on_response_error = kwargs.pop('raise_on_response_error', True)
```

Ajoutons plutôt un explicite raise_on_response_error=True dans la fonction.

Par ailleurs mon expérience avec nos amies les webservices me ferait ajouter une vérification que result est un dictionnaire, avant de faire le result.get('message').

#13 - 05 octobre 2022 13:10 - Valentin Deniaud

Perso je pense qu'il faut revenir en arrière, on a deux endpoints qui ne sont pas homogènes et n'ont pas la même manière d'indiquer une erreur, il ne faut pas chercher à mutualiser le code de gestion d'erreur dans self.request. Là ce code on se retrouve à l'utiliser pour un endpoint et le désactiver pour l'autre, sa place est donc directement dans self.update_credit_left (c'est revenir sur l'idée de Thomas, à voir ce qu'il en dit).

PS : ajouter un test pour couvrir le nouveau code c'est toujours mieux (j'aurais pu dire ça avant de valider tout à l'heure), bonus ça rendrait explicite à la lecture du patch l'erreur qu'on est en train de gérer (« le token est pas bon »).

#14 - 06 octobre 2022 01:01 - Thomas Noël

Valentin Deniaud a écrit :

Perso je pense qu'il faut revenir en arrière, on a deux endpoints qui ne sont pas homogènes et n'ont pas la même manière d'indiquer une erreur, il ne faut pas chercher à mutualiser le code de gestion d'erreur dans self.request. Là ce code on se retrouve à l'utiliser pour un endpoint et le désactiver pour l'autre, sa place est donc directement dans self.update_credit_left (c'est revenir sur l'idée de Thomas, à voir ce qu'il en dit).

Là, comme ça, rien. En fait je ne comprends finalement pas vraiment pourquoi il y a un "raise_on_response_error", dans ma tête self.request devrait toujours planter quand une erreur de token est détectée. Mais si c'est pas possible alors le faire uniquement pour self.update_credit_left oui, pourquoi pas. Mais c'est bizarre non ? (je dois rater un truc, as usual)

#15 - 06 octobre 2022 10:13 - Valentin Deniaud

Thomas Noël a écrit :

Là, comme ça, rien. En fait je ne comprends finalement pas vraiment pourquoi il y a un "raise_on_response_error", dans ma tête self.request devrait toujours planter quand une erreur de token est détectée.

On a 0 garantie que ce soit une erreur de token (message != 'OK' ça couvre toutes les erreurs).
Or il n'y a pas d'endpoint qui permette d'envoyer un SMS à plusieurs destinataires. Donc dans self.send_msg il y a une boucle qui self.request('/send/', un_numéro).
Au cas où une erreur soit retournée pour un numéro et pas pour les autres, le code essaye d'abord tous les numéros, puis remonte les erreurs. Si on ne met pas raise_on_response_error=False, alors on essaye plus tous les numéros, on plante à la première erreur.

(c'est juste un résumé de la situation, je ne sais pas comment bien gérer cette histoire qui peut aboutir à des envois partiels ingérables, mais c'est la faute de l'API et on sort du cadre de ce ticket)

le faire uniquement pour self.update_credit_left oui, pourquoi pas. Mais c'est bizarre non ?

C'est l'API qui est bizarre, j'insiste sur

on a deux endpoints qui ne sont pas homogènes et n'ont pas la même manière d'indiquer une erreur

Car d'un côté /send/ renvoie une clé status en plus de la clé message, de l'autre /credits/ renvoie juste message. (fun fact, quand il n'y a pas d'erreur status vaut 1, on a donc affaire à une API qualité platine)

#16 - 06 octobre 2022 12:15 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Valentin Deniaud a écrit :

Thomas Noël a écrit :

Là, comme ça, rien. En fait je ne comprends finalement pas vraiment pourquoi il y a un "raise_on_response_error", dans ma tête self.request devrait toujours planter quand une erreur de token est détectée.

On a 0 garantie que ce soit une erreur de token (message != 'OK' ça couvre toutes les erreurs).

J'avoue ici avoir parlé sans savoir, j'imaginai que l'erreur de token était identifiable par un moyen ou un autre (en fait leur API pourrait répondre une 403 mais apparemment c'est effectivement qualité platine).

Sans doute est-il utile donc de faire au plus bête et revenir au patch original.

Je (re)valide donc ce ticket MAIS pour le patch de la note 3 (qui avait été validé par Valentin).

#17 - 06 octobre 2022 14:59 - A. Berriot

- Fichier 0001-smsfactor-fixed-crash-when-updating-credit-69886.patch ajouté

- Statut changé de Solution validée à Solution proposée

#18 - 06 octobre 2022 14:59 - A. Berriot

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

```
commit 78df755c131fbac8fc4e43b9181e27319f194976
Author: Agate Berriot <aberriot@entrouvert.com>
Date: Thu Oct 6 14:47:14 2022 +0200
```

```
smsfactor: fixed crash when updating credit (#69886)
```

#19 - 07 octobre 2022 01:14 - Transition automatique

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

#20 - 11 décembre 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-smsfactor-fixed-crash-when-updating-credit-69886.patch	1,13 ko	05 octobre 2022	A. Berriot
0001-smsfactor-fixed-crash-when-updating-credit-69886.patch	932 octets	05 octobre 2022	A. Berriot
0001-smsfactor-fixed-crash-when-updating-credit-69886.patch	927 octets	05 octobre 2022	A. Berriot
0001-smsfactor-fixed-crash-when-updating-credit-69886.patch	2,73 ko	05 octobre 2022	A. Berriot

