

## Passerelle - Development #42230

### Uniformiser les retours des connecteurs SMS

28 avril 2020 19:26 - Nicolas Roche

<b>Statut:</b>	Rejeté	<b>Début:</b>	28 avril 2020
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>		<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
... en vue de gérer plusieurs tentatives sur l'envoi asynchrone des SMS en cas d'erreurs récupérables.  Les erreurs récupérables seront à priori les erreur de transport HTTP qui lèvent une RequestException, mais pourraient aussi être fonction de la réponse du service. Par exemple, pour les service qui retourne cette information, on pourrait tenter de renvoyer les SMS qui ne peuvent pas être délivrés, faute de crédit insuffisant sur le compte du client chez le fournisseur de service.			
<b>Demandes liées:</b>			
Lié à Passerelle - Development #21465: connecteurs SMS : file pour envois asy...		<b>Fermé</b>	<b>29 janvier 2018</b>
Lié à Passerelle - Bug #44730: Erreur SMS OVH : Too much requests. Please ret...		<b>Fermé</b>	<b>02 juillet 2020</b>

#### Historique

##### #1 - 28 avril 2020 19:27 - Nicolas Roche

- Lié à Development #21465: connecteurs SMS : file pour envois asynchrones ajouté

##### #2 - 29 avril 2020 15:09 - Nicolas Roche

- Fichier 0001-sms-raise-on-recoverable-errors-42230.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Patch qui crée une exception dédiée aux erreurs récupérables.

##### #3 - 25 mai 2020 10:13 - Frédéric Péters

Ça m'a l'air de vouloir introduire un truc générique et je dirais non, particulièrement quand ça part du cas particulier des SMS où les envois vont être asynchrones et où donc jamais il n'y aura un retour d'API avec ça.

Je dirais que l'ordre des choses devrait plutôt être d'avoir l'envoi asynchrone, et une fois ça en place alors on pourra s'interroger sur les possibilités de rejeu.

##### #4 - 10 juillet 2020 16:13 - Nicolas Roche

- Lié à Bug #44730: Erreur SMS OVH : Too much requests. Please retry in 3 seconds. ajouté

##### #5 - 02 septembre 2021 12:23 - Nicolas Roche

- Statut changé de Solution proposée à Nouveau
- Assigné à Nicolas Roche supprimé

A présent on a l'envoi asynchrone, mais on n'a pas encore été trop embêté par le rejeu.

S'il fallait traiter les erreur réseau, alors les attraper pour lever SkipJob permettrait le rejeu (à condition de plus traiter cette exception en amont).

ex:

```
def send_job(self, *args, **kwargs):
    try:
        self.send_msg(**kwargs)
    except requests.RequestException as exc:
```

```
raise SkipJob()
```

Vu l'unique moment où l'on a constaté que le rejeu nous serait utile (<https://dev.entrouvert.org/issues/44730#note-12>), il pourrait être utile de gérer une exception générique comme ce fut proposé dans ce ticket.  
ex:

```
error running send_job job (OVH error: {'status': 429, 'message': 'Too much requests.\nPlease retry in 3 seconds.\n'})
```

#### #6 - 02 septembre 2021 13:31 - Benjamin Dauvergne

On a déjà SkipJob pour ça, c'est le boulot du job de convertir les exceptions quelconques en SkipJob; par contre rien n'empêche de sous-classer APIError dans le connecteur pour pouvoir les différencier plus facilement dans le job; à la rigueur je vois bien une TransportAPIError pour tout ce qui est erreur au niveau TCP, mais là encore au niveau requests il faudrait différencier une erreur SSL qui n'est pas récupérable en général d'une erreur de connection ou de timeout.

Par contre je verrai bien une amélioration au constructeur de SkipJob pour pouvoir écrire SkipJob(delay=timedelta(minutes=10)) pour dire de pas réessayer 10 minutes, plutôt que de faire des calculs de timestamp.

#### #7 - 02 septembre 2021 14:39 - Nicolas Roche

c'est le boulot du job de convertir les exceptions quelconques en SkipJob

Oui, on ne fait plus que l'asynchrone, c'est vrai.  
(du coup je vais pouvoir rejeter ce ticket)

Par contre je verrai bien une amélioration au constructeur de SkipJob pour pouvoir écrire SkipJob(delay=timedelta(minutes=10)) pour dire de pas réessayer 10 minutes, plutôt que de faire des calculs de timestamp.

Sur MDEL\_DDPACS je vois que SkipJob est utilisée avec le paramètre after\_timestamp, qui semble gérer à la fois une date donnée mais également un delta.

```
raise SkipJob(after_timestamp=3600 * 6)
```

#### #8 - 02 septembre 2021 15:57 - Benjamin Dauvergne

Nicolas Roche a écrit :

Par contre je verrai bien une amélioration au constructeur de SkipJob pour pouvoir écrire SkipJob(delay=timedelta(minutes=10)) pour dire de pas réessayer 10 minutes, plutôt que de faire des calculs de timestamp.

Sur MDEL\_DDPACS je vois que SkipJob est utilisée avec le paramètre after\_timestamp, qui semble gérer à la fois une date donnée mais également un delta.  
[...]

Ah ben oui c'est déjà géré, par du code que j'ai écrit en plus, formidable.

#### #9 - 02 septembre 2021 16:23 - Nicolas Roche

- Statut changé de Nouveau à Rejeté

c'est le boulot du job de convertir les exceptions quelconques en SkipJob

#### Fichiers

---

0001-sms-raise-on-recoverable-errors-42230.patch	12,2 ko	29 avril 2020	Nicolas Roche
--	---------	---------------	---------------