

Passerelle - Bug #55840

Pouvoir tenter un envoi synchrone avant de lancer le job

26 juillet 2021 17:02 - Nicolas Roche

Statut:	Rejeté	Début:	26 juillet 2021
Priorité:	Normal	Echéance:	
Assigné à:	Nicolas Roche	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
L'idée étant qu'on fait du synchrone si tout va bien, et sinon on continue en asynchrone (ce serait bien que ce soit une option de add_job).			
Ébauche de patch ici : https://dev.entrouvert.org/issues/55230?issue_count=219&issue_position=11&next_issue_id=54663&prev_issue_id=55284#note-27			
Demandes liées:			
Lié à Passerelle - Development #55230: toulouse_smart: endpoint POST Interven...		Fermé	28 juin 2021

Historique

#1 - 26 juillet 2021 17:41 - Nicolas Roche

- Fichier 0001-jobs-add-a-try_now-parameter-to-run-new-jobs-synchro.patch ajouté
- Tracker changé de Support à Bug
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

#2 - 26 juillet 2021 17:42 - Nicolas Roche

- Lié à Development #55230: toulouse_smart: endpoint POST Intervention ajouté

#3 - 26 juillet 2021 19:12 - Thomas Noël

On peut pas profiter du transaction.on_commit qui est déjà là, et faire juste :

```
- transaction.on_commit(lambda: job.run(spool=True))
+ transaction.on_commit(lambda: job.run(spool=not try_now))
```

#4 - 27 juillet 2021 09:36 - Benjamin Dauvergne

Thomas Noël a écrit :

On peut pas profiter du transaction.on_commit qui est déjà là, et faire juste :

[...]

Si on est vraiment dans une transaction ça ne va pas faire ce qu'on espère, c'est à dire lancer le job immédiatement histoire d'avoir son résultat dans job.status, on aura systématiquement job.status == 'registered' et rien d'autre.

#5 - 27 juillet 2021 09:56 - Thomas Noël

Ok.

Maintenant pourquoi un "except SkipJob", vu que c'est déjà pris en charge dans le run ?

Aussi, si le try_now a réussi (job.status == 'completed'), je serais pour ne pas continuer (ne pas faire le transaction.on_commit(lambda: job.run(spool=True))) et juste faire un return job.

Autrement dit je pense à :

```
if try_now:
    job.run(spool=False)
```

```
if job.status == 'completed':  
    return job
```

#6 - 27 juillet 2021 10:09 - Benjamin Dauvergne

Thomas Noël a écrit :

Autrement dit je pense à :
[...]

Si tu veux mais il faut surtout avoir conscience que dans le spooler ou dans le job CRON, on peut avoir un timeout aussi long qu'on veut, c'est pour ça qu'on s'en sert, mais lors de l'appel synchrone il faut faire attention voir forcer un timeout court ou plus court; tout l'intérêt c'est d'avoir une réponse quasi immédiate quand ça marche, sinon de se reposer sur une exécution asynchrone sans trop de délais.

Ça c'est pour l'aspect timeout.

Coté transactionnel, quand on utilise try_now il faut éviter d'être dans une transaction ou bien s'assurer que ce qu'on fait dans le job est transactionnel, i.e. peut-être rejoué sans souci si la transaction autour échoue.

#7 - 27 juillet 2021 11:22 - Thomas Noël

Benjamin Dauvergne a écrit :

Thomas Noël a écrit :

Autrement dit je pense à :
[...]

Si tu veux mais il faut surtout avoir conscience que dans le spooler ou dans le job CRON, on peut avoir un timeout aussi long qu'on veut, c'est pour ça qu'on s'en sert, mais lors de l'appel synchrone il faut faire attention voir forcer un timeout court ou plus court; tout l'intérêt c'est d'avoir une réponse quasi immédiate quand ça marche, sinon de se reposer sur une exécution asynchrone sans trop de délais.

Ça c'est pour l'aspect timeout.

Coté transactionnel, quand on utilise try_now il faut éviter d'être dans une transaction ou bien s'assurer que ce qu'on fait dans le job est transactionnel, i.e. peut-être rejoué sans souci si la transaction autour échoue.

Parfaitement d'accord avec tout ça. L'idée du try_now n'est pas de moi :)

Pour tout dire au départ j'étais contre l'idée présentée dans ce ticket. Mais il semble que ça soit utile pour SMART ? Si ce n'est pas le cas, je préfère ne pas avoir d'implémentation de cette affaire qui peut donner de faux espoirs.

Quand le besoin de "tenter d'abord et passer en job sinon", je préférerais que l'appel synchrone soit fait directement par l'appelant avant de basculer en job en cas de pépin (timeout rapide ou autre).

#8 - 27 juillet 2021 11:24 - Frédéric Péters

Pour info pareil pas fan de l'idée, surtout parce que ça va amener des workflows qui vont être faits en comptant sur les infos d'un résultat immédiat alors que celui-ci ne sera pas garanti.

#9 - 27 juillet 2021 17:54 - Nicolas Roche

- Statut changé de Solution proposée à Rejeté

Je rejette.

A la base c'est parti d'un faux problème qui était de retrouver les erreurs synchrones/asynchrones au même endroit.

En fait il suffit juste de les recopier.

<https://dev.entrouvert.org/issues/55230#note-26>

Fichiers

0001-jobs-add-a-try_now-parameter-to-run-new-jobs-synchro.patch	3,04 ko	26 juillet 2021	Nicolas Roche
-----------------------------------------------------------------	---------	-----------------	---------------