

## Passerelle - Development #67613

### SMS envoyés en double (via OVH ?)

21 juillet 2022 16:09 - Thomas Noël

<b>Statut:</b>	Fermé	<b>Début:</b>	21 juillet 2022
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Valentin Deniaud	<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		

**Description**

Déjà vu deux fois chez deux clients : des SMS envoyés une seule fois dans le workflow, mais reçu deux fois par l'utilisateur.

Quand on regarde au niveau de passerelle, on voit que cela correspond à des "job" dont les moments de création et d'exécution sont séparés de quelques instants... Le premier SMS reçu est au moment de la création du job, le second au moment de l'exécution.

Difficile d'en dire plus, on manque un peu de logs sur les jobs. Mais il faudrait creuser, le nombre de SMS en double se compte en milliers.

#### Révisions associées

##### Révision 2d6e8599 - 25 juillet 2022 13:50 - Valentin Deniaud

jobs: add missing checks in runjob command (#67613)

##### Révision 8ae6e9e4 - 25 juillet 2022 14:06 - Valentin Deniaud

jobs: add missing checks in runjob command (#67613)

#### Historique

##### #4 - 25 juillet 2022 13:28 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

Vu avec Thomas, on est probablement dans le cas où le cron va jouer dans BaseResource.jobs :

```
584     def jobs(self):
...
605         with transaction.atomic():
606             job = (
607                 self.jobs_set()
608                 .exclude(pk__in=skipped_jobs)
609                 .filter(
610                     Q(after_timestamp__isnull=True) | Q(after_timestamp__lt=timzone.now()),
611                     status='registered',
612                 )
613                 .select_for_update(**skip_locked)
614                 .order_by('pk')[0]
615                 .first()
616             )
617             if not job:
618                 break
619             job.status = 'running'
620             job.save()
621             # release lock
622             result = job.run()
```

En même temps, exécution de la commande runjob par le spooler :

```
29     def handle(self, *args, **options):
...
33         with transaction.atomic():
34             try:
35                 job = Job.objects.select_for_update(**skip_locked).get(pk=options['job_id'])
```

```
36         except Job.DoesNotExist:
37             raise CommandError('missing job')
38         job.status = 'running'
39         job.save()
40         # release lock
41         job.run()
```

Le code se protège des race conditions de la manière suivante :

- Dans BaseResource.jobs, récupération du job
- Il est locké par le select\_for\_update
- Si il y a un job en même temps, le skip\_locked fait que le job en cours de traitement n'apparaît pas

Mais, gros trou dans la raquette dans le scénario où :

- Dans BaseResource.jobs, récupération du job
- Il est locké par le select\_for\_update
- On le place en statut « running »
- On enlève le lock
- On l'exécute

Il peut se produire la race condition :

- Dans BaseResource.jobs, récupération du job
- Il est locké par le select\_for\_update
- On le place en statut « running »
- On enlève le lock
- Un runjob parallèle récupère le job (car plus de lock)
- Exécution par runjob et par BaseResource.jobs

En fait c'est jusque le .filter(status='registered') manque côté runjob (en profiter aussi pour ajouter celui sur le timestamp).

#### #5 - 25 juillet 2022 13:35 - Thomas Noël

Valentin Deniaud a écrit :

En fait c'est jusque le .filter(status='registered') manque côté runjob (en profiter aussi pour ajouter celui sur le timestamp).

Même analyse, mais plutôt que des filter, faire :

```
if job.status != 'registered':
    raise CommandError('cannot run job, status is %s' % job.status)
if job.after_timestamp and job.after_timestamp > maintenant:
    raise CommandError('cannot run job, should be run after ...
...

```

pour avoir des messages de refus plus clairs ?

#### #6 - 25 juillet 2022 13:51 - Valentin Deniaud

- Fichier 0001-jobs-add-missing-checks-in-runjob-command-67613.patch ajouté
- Tracker changé de Support à Development
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

#### #7 - 25 juillet 2022 14:01 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

#### #8 - 25 juillet 2022 14:02 - Valentin Deniaud

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

```
commit 2d6e859971b76a3b1ffed8addb513e102a9d83a4
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Mon Jul 25 13:42:48 2022 +0200

```

```
jobs: add missing checks in runjob command (#67613)
```

#### #11 - 26 juillet 2022 01:39 - Thomas Noël

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

Déployé aussi en hotfix. Activité à surveiller ce mardi.

#### #12 - 25 septembre 2022 04:42 - Transition automatique

Automatic expiration

#### Fichiers

---

0001-jobs-add-missing-checks-in-runjob-command-67613.patch	2,61 ko	25 juillet 2022	Valentin Deniaud
------------------------------------------------------------	---------	-----------------	------------------