

## w.c.s. - Bug #24078

### confusion connexion sql en afterjob (?)

25 mai 2018 09:23 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	25 mai 2018
<b>Priorité:</b>	Haut	<b>Echéance:</b>	
<b>Assigné à:</b>	Frédéric Péters	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	
<b>Patch proposed:</b>	Oui		

**Description**

```
Exception:
  type = '<class 'psycpg2.ProgrammingError'>', value = 'relation "formdata_17_rdv_evolution" does not exist
LINE 2: ...                                comment, parts FROM formdata_1...
                                     ^
'

Stack trace (most recent call first):
  File "/usr/lib/python2.7/dist-packages/wcs/sql.py", line 1218, in get_evolution
    1216         WHERE formdata_id = %(id)s
    1217         ORDER BY id''' % self._table_name
> 1218     cur.execute(sql_statement, {'id': self.id})
    1219     self._evolution = []
    1220     while True:
```

Associé à un vhost qui ne correspond pas au formdef en question...

### Révisions associées

#### Révision 93374961 - 25 mai 2018 12:07 - Frédéric Péters

threads: make sure logger and substitution singletons are not reused (#24078)

### Historique

#### #1 - 25 mai 2018 09:27 - Frédéric Péters

- Fichier 0001-afterjobs-make-sure-postgresql-connection-is-not-reu.patch ajouté
- Statut changé de Nouveau à En cours
- Patch proposed changé de Non à Oui

(shooting in the dark)

#### #2 - 25 mai 2018 11:05 - Frédéric Péters

- Fichier 0001-threads-make-sure-logger-and-substitution-singletons.patch ajouté

Meilleure analyse.

On fait pour un nouveau thread un `copy.copy()` du publisher mais des objets à l'intérieur restent partagés.

Faire un `deepcopy` n'est pas possible parce que ça creuse trop loin et échoue à un moment sur des objets qui ne peuvent être copiés, du coup plutôt une méthode `reset()` qui assure pour le logger (parce que) et le substitutions (le bug ici) qu'un nouvel objet sera utilisé.

#### #3 - 25 mai 2018 11:45 - Thomas Noël

Ça semble juste.

Peut-être que le `reset()` pourrait s'appeler `detach()` ou autre, ça semble bizarre de faire un `reset()` après un `copy`... En fait je me dis qu'on pourrait avoir une méthode `copy(self)` dans `CommonPublisher`, qui ferait le `copy.copy` et ces opérations de `reset`. C'est vraiment juste une question de "lisibilité", Ack si tu penses que c'est assez lisible ainsi.

#### #4 - 25 mai 2018 12:08 - Frédéric Péters

- Statut changé de *En cours à Résolu* (à déployer)

Poussé avec `.detach()`. (je trouve ça plus clair que caché derrière un copy).

```
commit 93374961c2acba3341559a25cfb6e5e5a6c2c62e
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Fri May 25 09:25:46 2018 +0200
```

```
threads: make sure logger and substitution singletons are not reused (#24078)
```

#### #5 - 23 décembre 2018 14:39 - Frédéric Péters

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

#### Fichiers

0001-afterjobs-make-sure-postgresql-connection-is-not-reu.patch	891 octets	25 mai 2018	Frédéric Péters
0001-threads-make-sure-logger-and-substitution-singletons.patch	1,45 ko	25 mai 2018	Frédéric Péters