

## Hobo - Development #32685

### multitenant: les classes Thread surchargées doivent être chargées le plus tôt possible

29 avril 2019 18:30 - Benjamin Dauvergne

<b>Statut:</b>	Fermé	<b>Début:</b>	29 avril 2019
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Benjamin Dauvergne	<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>			
Pour éviter à d'autres librairies (exemple, paramiko) d'utiliser les deux il faut charger celles-ci le plus tôt possible, voir <a href="#">#32683</a> .			
<b>Demandes liées:</b>			
Lié à Passerelle - Support #32683: sftp: incompatibilité entre paramiko et la...		<b>Rejeté</b>	<b>29 avril 2019</b>

#### Révisions associées

##### Révision 01a24f29 - 30 avril 2019 18:19 - Benjamin Dauvergne

multitenant: load multitenant thread classes early on (#32685)

Also do not keep a tenant around if current database wrapper is not multitenant aware.

#### Historique

##### #1 - 29 avril 2019 18:36 - Benjamin Dauvergne

- Lié à Support #32683: sftp: incompatibilité entre paramiko et la surcharge de threading par hobo ajouté

##### #2 - 29 avril 2019 18:38 - Benjamin Dauvergne

- Fichier 0001-multitenant-load-multitenant-thread-classes-early-on.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Ça devrait faire l'affaire.

##### #3 - 30 avril 2019 11:35 - Christophe Siraut

Pour éviter à d'autres librairies (exemple, paramiko) d'utiliser les deux il faut charger celles-ci le plus tôt possible

les deux quoi? Le code déplacé est identique, j'imagine que l'import en début de apps.py fait la différence.

##### #4 - 30 avril 2019 12:01 - Benjamin Dauvergne

Christophe Siraut a écrit :

Pour éviter à d'autres librairies (exemple, paramiko) d'utiliser les deux il faut charger celles-ci le plus tôt possible

les deux quoi? Le code déplacé est identique, j'imagine que l'import en début de apps.py fait la différence.

Pardon d'utiliser les deux implémentations des threads, le fait est que paramiko est mal écrit, il fait directement appel à Thread.init() au lieu d'utiliser super() et ça casse tout (il appelle TenantAwareThread.\_\_init\_\_() alors que sa classe n'en hérite pas).

À terme il faudrait arrêter avec le monkeypatching et utiliser directement TenantAwareThread quand c'est nécessaire.

##### #5 - 30 avril 2019 13:10 - Benjamin Dauvergne

- Fichier 0001-multitenant-load-multitenant-thread-classes-early-on.patch ajouté

Typo idiote, 'False' -> False.

#### #6 - 30 avril 2019 15:05 - Benjamin Dauvergne

- Fichier 0001-multitenant-load-multitenant-thread-classes-early-on.patch ajouté

Maintenant il faut prendre en compte le fait de ne pas tourner dans un environnement multitenant (entre les tests).

#### #7 - 30 avril 2019 18:01 - Nicolas Roche

- Statut changé de Solution proposée à Solution validée

On se place au début des chargements (quand hobo.multitenant est chargé).  
eomme passerelle.util.sftp ne sera appelé que par des applications passerelle bien après, ça ira :

```
hobo/multitenant/_init__.py:_
```

```
...  
threads.install_tenant_aware_threads()
```

surcharge /usr/lib/python2.7/threading.py.

Difficile à tester car dans un test on est forcément après AppConfig.ready().  
Là on essaye de se placer avant AppConfig.ready() :  
avant le chargement de l'ensemble des applications Django, sachant que paramiko est chargé à ce moment.

Le code n'est pas exactement pareil à cause des tests de hobo qui ne tournent pas en multitenant : si le driver de DB n'est pas celui de django-tenant-schemas connection.get\_tenant() n'existe pas.  
Or en déplaçant l'initialisation plus tôt on constate qu'un truc quelque part charge hobo.multitenant.

Merci pour les explications. Je valide.

#### #8 - 30 avril 2019 18:20 - Benjamin Dauvergne

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

```
commit 01a24f29a59322c9f1b83e36cdf5f5e20a156e9c  
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>  
Date: Mon Apr 29 18:35:57 2019 +0200
```

```
multitenant: load multitenant thread classes early on (#32685)
```

```
Also do not keep a tenant around if current database wrapper is not  
multitenant aware.
```

#### #9 - 30 avril 2019 20:15 - Frédéric Péters

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

#### Fichiers

0001-multitenant-load-multitenant-thread-classes-early-on.patch	10,2 ko	29 avril 2019	Benjamin Dauvergne
0001-multitenant-load-multitenant-thread-classes-early-on.patch	10,2 ko	30 avril 2019	Benjamin Dauvergne
0001-multitenant-load-multitenant-thread-classes-early-on.patch	10,5 ko	30 avril 2019	Benjamin Dauvergne