

## Passerelle - Bug #5593

### Refonte des settings

26 septembre 2014 15:23 - Benjamin Dauvergne

<b>Statut:</b>	Fermé	<b>Début:</b>	26 septembre 2014
<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>	
<b>Patch proposed:</b>	Oui		

**Description**

Le premier but est de pouvoir désigner simplement sur la ligne de commande le fichier settings à charger sans avoir de problème avec les chemins d'import. Le deuxième but est de concentrer la spécialisation pour le packaging dans le packaging Debian et pas dans le projet python. Le dernier but c'est de limiter au maximum ce qui doit s'exécuter après le chargement de la configuration locale (toutes les conditionnelles qu'on pouvait trouver après le import local\_settings) et n'est donc pas surchargeable.

Le premier patch concerne le projet Python, il éclate le fichier settings en trois:

- default\_settings.py: toutes les valeurs par défaut nécessaires à Passerelle en enlevant celle concernant l'installation sous Debian et/ou celle existant déjà dans django.conf.globalsettings
- settings.py: charge default\_settings.py, et tente d'exécuter le fichier désigné par la variable d'environnement DJANGO\_CONFIG\_FILE
- tenant\_settings.py: idem que pour settings.py, mais ajoute aux settings par défaut ce qu'il faut pour le multitenant (et se plaint si python-entrouvert n'est pas installé)

Il modifie aussi passerelle\_manage.py pour y ajouter deux flags à mettre avant la commande Django:

- -c config.py permet de désigner le fichier de configuration, c'est obligatoire,
- -t active le mode multitenant en chargeant tenant\_settings.py au lieu de settings.py.

Le deuxième patch modifie le packaging Debian en ajoutant un fichier /usr/lib/passerelle/debian\_config.py qui définit les chemins spécifiques à Debian et puis charge si il est présent /etc/passerelle/passerelle.conf.

Le but est de rendre le chargement des settings purement linéaire. Si à terme un projet souhaite définir un format supplémentaire en plus de Python pour son fichier de settings il pourra (à détecter dans [tenant\_]settings.py lors du chargement).

### Révisions associées

#### Révision 5dbf5387 - 02 octobre 2014 09:54 - Benjamin Dauvergne

New organization of settings

Only one setting is now available from the environment: DJANGO\_CONFIG\_FILE, it gives a path to a Python that will be executed at the end of the settings module.

For simpler usage this environment is set by passerelle\_manage.py by passing it a new mandatory option '--config=/path/to/config'.

Tenant settings are now activated by passing the '--multitenant' flag to passerelle\_manage.py.

All OS platform specific paths have been removed, it's expected that packaging will add them. Debian for example would provide a debian\_config.py file to adapt to its own paths.

#### Révision 85c73a35 - 02 octobre 2014 13:49 - Thomas Noël

Adapt to new settings organization (#5593)

### Historique

#### #1 - 26 septembre 2014 16:12 - Jérôme Schneider

On en a parlé à midi par oral (je ne fais que le remettre mon avis par écrit ici). J'aime bien l'idée de supprimer la magie et de rajouter un fichier de configuration avec une option. Ce qui me gêne c'est le fait d'avoir un fichier de configuration en Python qui ouvre la voie à de la bidouille et de la non

lisibilité dans ce fichier. Autant ça ne me gêne pas pour des fichiers auto générés autant là ça me semble plus problématique.

J'ai déjà eu comme réponse qu'il faudra râler dans ce cas là et demander au responsable de nettoyer la conf. Je continue de penser que d'avoir une conf propre dès le départ c'est mieux.

## #2 - 26 septembre 2014 16:29 - Thomas Noël

Pour ma part pas de soucis pour avoir un execfile pour lire la config, transformer un fichier .ini me parait complexe (et ajoutera de la magie par rapport à un fichier python de base avec juste des TRUC=valeur)

En première lecture :

- dans le manage, je ferais un `config_file=os.environ.get('DJANGO_CONFIG_FILE', False)` histoire d'être parallèle avec la gestion des choses dans `wsgi.py` (ce dernier ne pouvant se baser que sur des variables d'env)
- cosmétique, des options auto-documentées : `--multitenant` au lieu de `-t` et `--config=/truc` au lieu de `-c truc` (ce qui devrait être plus simple à gérer, en plus)
- virer le `"from django.conf import global_settings"` au début de `tenant_settings.py`
- livrer `debian_config.py` dans le master, c'est un exemple de configuration pour le déploiement (i.e. ne pas le cacher dans le paquet debian)
- peut-être faire en sorte de `tenant_settings.py` se base sur `settings.py` ?

## #3 - 30 septembre 2014 16:36 - Thomas Noël

- Fichier `0001-New-organization-of-settings-VersionDeThomas.patch` ajouté

Voilà une version relue et un peu revue par mes soins (et donc surtout, les options renommées `--config=...` et `--multitenant`, parce que j'aime ça les double moins)

## #4 - 30 septembre 2014 17:07 - Thomas Noël

Pour la partie Debian, je verrais les choses dans un autre ordre :

- un passerelle/`debian_config.py` livré par l'application (dans le master du projet)
- faire un export `DJANGO_CONFIG_FILE=/etc/passerelle/config.py` dans le `init.d`

Et livrer un `/etc/passerelle/config.py` par défaut :

```
# /etc/passerelle/config.py

from passerelle.debian_config import *

# add local configuration bellow
#TIME_ZONE = 'Europe/Paris'
#LANGUAGE_CODE = 'fr-fr'
```

Ça marcherait ?

Ajouter aussi un `/etc/default/passerelle` :

```
DJANGO_SETTINGS_MODULE=passerelle.settings
# if you want to use multitenant mode, set :
#DJANGO_SETTINGS_MODULE=passerelle.tenant_settings
```

Notes avant que je monte dans le train :

- ajouter `TENANT_BASE=/var/lib/passerelle/tenants/ (?)` dans `debian_config.py`
- je pense qu'il manque un export `DJANGO_CONFIG_FILE` dans `init.d`, ou pas, voilà, le RER m'attend

## #5 - 02 octobre 2014 13:55 - Thomas Noël

Thomas Noël a écrit :

Pour la partie Debian, je verrais les choses dans un autre ordre :

- un passerelle/`debian_config.py` livré par l'application (dans le master du projet)
- faire un export `DJANGO_CONFIG_FILE=/etc/passerelle/config.py` dans le `init.d`

Finalement j'ai gardé la technique de Benjamin (`debian_config.py` qui exec `config.py`).

## #6 - 10 octobre 2014 16:42 - Benjamin Dauvergne

- Statut changé de Nouveau à Fermé

C'est commité, non ?

```
commit 5dbf5387b8bb58050d8135bdd94ceb9ebbcd45da
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Thu Oct 2 09:54:39 2014 +0200
```

New organization of settings

Only one setting is now available from the environment: `DJANGO_CONFIG_FILE`, it gives a path to a Python that will be executed at the end of the settings module.

For simpler usage this environment is set by `passerelle_manage.py` by passing it a new mandatory option `'--config=/path/to/config'`.

Tenant settings are now activated by passing the `'--multitenant'` flag to `passerelle_manage.py`.

All OS platform specific paths have been removed, it's expected that packaging will add them. Debian for example would provide a `debian_config.py` file to adapt to its own paths.

Je ferme.

## Fichiers

---

0001-New-organization-of-settings.patch	20,1 ko	26 septembre 2014	Benjamin Dauvergne
0001-Adapt-to-new-settings-organization.patch	4,06 ko	26 septembre 2014	Benjamin Dauvergne
0001-New-organization-of-settings-VersionDeThomas.patch	20,1 ko	30 septembre 2014	Thomas Noël