Hobo - Bug #34055

ne pas crasher dans un settings.json n'est pas du json valide

17 juin 2019 15:23 - Thomas Noël

Statut:	Nouveau	**Début:**	17 juin 2019
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

Crash complet du middleware multitenant quand un settings.json est mal formatté :

```
juin 17 15:18:51 combo uwsgi[29002]: Traceback (most recent call last):
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/dist-packages/django/core/handlers
/wsgi.py", line 154, in __call__
juin 17 15:18:51 combo uwsgi[29002]:     set_script_prefix(get_script_name(environ))
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/dist-packages/django/core/handlers
/wsgi.py", line 188, in get_script_name
juin 17 15:18:51 combo uwsgi[29002]:     if settings.FORCE_SCRIPT_NAME is not None:
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/dist-packages/hobo/multitenant/app
s.py", line 23, in <lambda>
juin 17 15:18:51 combo uwsgi[29002]:     conf.LazySettings.__getattr__ = lambda self, name: getatt
r(self._wrapped, name)
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/dist-packages/hobo/multitenant/set
tings.py", line 97, in __getattr__
juin 17 15:18:51 combo uwsgi[29002]:     return getattr(self.get_wrapped(), name)
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/dist-packages/hobo/multitenant/set
tings.py", line 92, in get_wrapped
juin 17 15:18:51 combo uwsgi[29002]:     return self.get_tenant_settings(tenant)
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/dist-packages/hobo/multitenant/set
tings.py", line 80, in get_tenant_settings
juin 17 15:18:51 combo uwsgi[29002]:     tenant_settings, last_time = self.load_tenant_settings(te
nant, tenant_settings, last_time)
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/dist-packages/hobo/multitenant/set
tings.py", line 66, in load_tenant_settings
juin 17 15:18:51 combo uwsgi[29002]:     loader.update_settings(tenant_settings, tenant)
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/dist-packages/hobo/multitenant/set
tings_loaders.py", line 32, in update_settings
juin 17 15:18:51 combo uwsgi[29002]:     self.update_settings_from_path(tenant_settings, path)
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/dist-packages/hobo/multitenant/set
tings_loaders.py", line 390, in update_settings_from_path
juin 17 15:18:51 combo uwsgi[29002]:     self.handle_settings(tenant_settings, json.load(f))
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/json/__init__.py", line 291, in lo
ad
juin 17 15:18:51 combo uwsgi[29002]:     **kw)
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/json/__init__.py", line 339, in lo
ads
juin 17 15:18:51 combo uwsgi[29002]:     return _default_decoder.decode(s)
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/json/decoder.py", line 364, in dec
ode
juin 17 15:18:51 combo uwsgi[29002]:     obj, end = self.raw_decode(s, idx=_w(s, 0).end())
juin 17 15:18:51 combo uwsgi[29002]:   File "/usr/lib/python2.7/json/decoder.py", line 380, in raw
_decode
juin 17 15:18:51 combo uwsgi[29002]:     obj, end = self.scan_once(s, idx)
juin 17 15:18:51 combo uwsgi[29002]: ValueError: Expecting , delimiter: line 2 column 22 (char 23)
```

Il faut voir à être un peu plus permissif lors du self.handle_settings(tenant_settings, json.load(f))

Demandes liées:

Lié à Hobo - Bug #48597: Trace settings	**Fermé**	**17 novembre 2020**

Historique

#1 - 17 juin 2019 16:13 - Thomas Noël

Je coince sur le test.

```
diff --git a/hobo/multitenant/settings_loaders.py b/hobo/multitenant/settings_loaders.py
index a355f21..1421c7e 100644
--- a/hobo/multitenant/settings_loaders.py
+++ b/hobo/multitenant/settings_loaders.py
@@ -387,7 +387,12 @@ class SettingsJSON(FileBaseSettingsLoader, SettingsDictUpdateMixin):

     def update_settings_from_path(self, tenant_settings, path):
         with open(path) as f:
-            self.handle_settings(tenant_settings, json.load(f))
+            try:
+                settings_json = json.load(f)
+            except ValueError:
+                pass
+            else:
+                self.handle_settings(tenant_settings, settings_json)

 class DictAdapter(dict):
diff --git a/tests_multitenant/test_settings.py b/tests_multitenant/test_settings.py
index 98ee238..8331d25 100644
--- a/tests_multitenant/test_settings.py
+++ b/tests_multitenant/test_settings.py
@@ -303,3 +303,20 @@ def test_tenant_json_settings_reload(tenants, settings, freezer):
         for tenant in tenants:
             with tenant_context(tenant):
                 assert django.conf.settings.EXTEND_ME == [1, 3]
+
+        # move 1 minute in the future
+        freezer.move_to(datetime.timedelta(seconds=60))
+
+        # update EXTEND_ME tenant value
+        for tenant in tenants:
+            with open(os.path.join(tenant.get_directory(), 'settings.json'), 'w') as fd:
+                json.dump({
+                    'EXTEND_ME': [99]
+                }, fd)
+
+        # move 1 minute in the future
+        freezer.move_to(datetime.timedelta(seconds=60))
+
+        for tenant in tenants:
+            with tenant_context(tenant):
+                assert django.conf.settings.EXTEND_ME == [99]
```

Le test ici n'est pas encore écrit jusqu'à tester un json mal formé, je chercher d'abord à changer la valeur de EXTEND_ME à 99 mais ça ne passe pas, je n'ai pas encore compris pourquoi.

```
            for tenant in tenants:
                with tenant_context(tenant):
>                   assert django.conf.settings.EXTEND_ME == [99]
E                   assert [1, 3] == [99]
E                     At index 0 diff: 1 != 99
E                     Left contains more items, first extra item: 3
E                     Use -v to get the full diff
```

Pas encore compris ce qui se passe, je me demande si c'est pas freezer.move_to(datetime.timedelta(seconds=60)) qui, en fait, ne fait pas son job ?... Un peu perdu je suis.

#2 - 17 juin 2019 18:23 - Benjamin Dauvergne

Je pense qu'ils ne faut pas "juste ne pas planter", il faut garder l'ancienne version des setttings le temps que ça remarche et balancer par ailleurs un gros mail d'erreur (mais alors là faut doser, parce qu'on peut facilement s'en prendre 1000)... parce qu'à chaque rechargement les settings sont nettoyés, et si on ne charge pas settings.json on a juste un truc vide (modulo tout ce que les autres settings-loader peuvent charger).

En gros faudrait chopper l'exception plus haut pendant le parcours des settings-loader, si un foire, on revient aux settings précédents on fait un logger.exception(.), et on note dans les settings qu'on l'a fait

```
settings.LOADER_ERROR = True
```

, au prochain coup si LOADER_ERROR est à True on ne log rien ou on peut mettre un settings.LOADER_ERROR_TIMESTAMP = time.time() et si

ça fait moins d'1h on ne log rien (enfin tu fais comme tu veux).

#3 - 17 juin 2019 18:24 - Benjamin Dauvergne

On se mangera juste autant de mails que de workers mais c'est mieux que 1000.

#4 - 17 juin 2019 18:28 - Benjamin Dauvergne

Pour ton souci de test, je dirai de tracer dans load_tenant_settings() je ne vois pas mieux.

#5 - 15 juillet 2023 10:52 - Frédéric Péters

- Lié à Bug #48597: Trace settings ajouté

#6 - 15 juillet 2023 10:52 - Frédéric Péters

- Sujet changé de ne pas crasher dans un settings.json n'est pas du json à ne pas crasher dans un settings.json n'est pas du json valide