

## Hobo - Bug #32886

### tests sur cook.py (côté serveur)

07 mai 2019 19:26 - Nicolas Roche

<b>Statut:</b>	Fermé	<b>Début:</b>	07 mai 2019
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Nicolas Roche	<b>% réalisé:</b>	0%
<b>Catégorie:</b>	serveur	<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
Ticket pour ajouter des tests unitaires sur <i>hobo/environment/management/commands/cook.py</i>			
<b>Demandes liées:</b>			
Lié à Hobo - Bug #33457: code redondant		<b>Fermé</b>	<b>27 mai 2019</b>
Lié à Hobo - Bug #33536: tests sur cook.py (correction)		<b>Fermé</b>	<b>29 mai 2019</b>

### Révisions associées

#### Révision 2afc250a - 27 mai 2019 19:37 - Nicolas Roche

cook: add unit tests for cook.py (#32886)

### Historique

#### #1 - 07 mai 2019 19:49 - Nicolas Roche

- Fichier *0001-cook-add-unit-tests-for-cook.py-32886.patch* ajouté
- Fichier *0002-cook-add-tests-using-tenants-for-cook.py-32886.patch* ajouté
- Tracker changé de Autre à Bug
- Sujet changé de tests unitaires sur cook.py à tests sur cook.py
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

voici pour la partie serveur (qui ne teste pas hobo-deploy)

- j'ai l'impression que rien n'est prévu si `set_idp()` est appelé sans aucun service Authentic de configuré. Faut-il corriger ?
- je ne trouve pas dans quel contexte la fonction `cook()` est appelée (peut-être qu'il s'agit de code mort). Elle lève une exception si `self.permissive` n'est pas déclarée, idem faut-il corriger ?

#### #2 - 07 mai 2019 21:56 - Benjamin Dauvergne

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

Nicolas Roche a écrit :

voici pour la partie serveur (qui ne teste pas hobo-deploy)

- j'ai l'impression que rien n'est prévu si `set_idp()` est appelé sans aucun service Authentic de configuré. Faut-il corriger ?

Indique plus précisément le problème et la correction que tu envisages.

- je ne trouve pas dans quel contexte la fonction `cook()` est appelée (peut-être qu'il s'agit de code mort). Elle lève une exception si `self.permissive` n'est pas déclarée, idem faut-il corriger ?

Le journal de git est en général une saine lecture :

```
commit c25b14cf2517b0bd587f74261f67e73d03831290
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Wed Jun 28 22:34:12 2017 +0200
```

```
cook: add cook step, to run "sub-"recipes (#15797)
```

```
{
  "cook": {
    "filename": "secondary-recipe.json"
  }
}
```

C'est donc pour lancer des sous-recettes dans un cadre multi-publik avec des hobos primaires et secondaires, je laisse Fred en dire plus mais je pense que ça devrait être testé aussi, voir [#15797](#).

### #3 - 07 mai 2019 21:57 - Benjamin Dauvergne

Pour des tests en grandeur nature je signale cela : <https://pypi.org/project/pytest-rabbitmq/>

### #4 - 08 mai 2019 19:15 - Nicolas Roche

- *Statut changé de Résolu (à déployer) à Information nécessaire*

J'ai raté cette piste, merci.

Je pensais attraper les exceptions standards pour lever à la place des 'CommandError':

- IndexError pour test\_set\_idp() utilisée sans Authentic configuré
- AttributeError pour cook() utilisée avec permissive non déclaré
- AttributeError pour set\_theme() utilisée sans que les thèmes soient installés

En fait, cette 3ème erreur fait que le test pseudo 'grandeur nature' ne passe pas sur Jenkins. Est-ce que cela reste envisageable de pousser un tel test ?

### #5 - 08 mai 2019 22:37 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Pour des tests en grandeur nature je signale cela : <https://pypi.org/project/pytest-rabbitmq/>

Ça ne représente aucune nécessité dans le cadre de ce ticket, je signale juste pour plus tard (voir pour ouvrir un nouveau ticket suite à celui-ci).

### #6 - 09 mai 2019 09:56 - Benjamin Dauvergne

Nicolas Roche a écrit :

J'ai raté cette piste, merci.

Je pensais attraper les exceptions standards pour lever à la place des 'CommandError':

- IndexError pour test\_set\_idp() utilisée sans Authentic configuré
- AttributeError pour cook() utilisée avec permissive non déclaré
- AttributeError pour set\_theme() utilisée sans que les thèmes soient installés

Je n'ai pas vraiment le contexte mais réutiliser les exceptions de la lib standard est une idée acceptable pour des petites fonctions se rapprochant d'une fonction de base genre émettre IndexError/KeyError sur une erreur dans get\_truc(i). Mais s'en servir pour propager des erreurs à travers un graphe d'appel c'est une mauvaise idée, il vaut mieux dans ce cas prévoir une exception spécifique.

En fait, cette 3ème erreur fait que le test pseudo 'grandeur nature' ne passe pas sur Jenkins. Est-ce que cela reste envisageable de pousser un tel test ?

Désolé je ne comprends pas vraiment là, poste une trace peut-être.

### #7 - 09 mai 2019 10:44 - Nicolas Roche

Merci benj, j'essaye d'être plus explicite :

Les 3 levées d'exceptions décrites ci-dessus seraient atteignables directement via la ligne de commande

```
$ hobo-manage cook recipe.json
```

Il ne s'agit donc pas tant de gérer les exceptions propagées dans les appels mais plutôt celles reçues dans le terminal de l'administrateur.

Cf les 2 'TODO' dans les 2 patchs ci-dessus,

et la trace dans jenkins :

[https://jenkins2.entrouvert.org/job/hobo-wip/job/wip%252F32886-add-tests-for-cook/1/testReport/junit/tests\\_schemas/test\\_cook/test\\_cook\\_ozwillo/](https://jenkins2.entrouvert.org/job/hobo-wip/job/wip%252F32886-add-tests-for-cook/1/testReport/junit/tests_schemas/test_cook/test_cook_ozwillo/)

La seconde question (erreur avec tox, désolé d'avoir embrouillé le ticket) c'est de savoir si c'est envisageable de modifier jenkins pour prendre en compte les thèmes. Je dis ça parce qu'il faut que les thèmes ne soient pas installés (via make install) pour reproduire la trace jenkins ci-dessus.

#### #8 - 09 mai 2019 13:08 - Benjamin Dauvergne

Nicolas Roche a écrit :

Merci benj, j'essaye d'être plus explicite :

Les 3 levées d'exceptions décrites ci-dessus seraient atteignables directement via la ligne de commande [...]

Oui oui mais donc on devrait avoir une méthode cook générique qui génère des CookError et la commande doit convertir ça en CommandError proprement.

Cf les 2 'TODO' dans les 2 patchs ci-dessus, et la trace dans jenkins :

[https://jenkins2.entrouvert.org/job/hobo-wip/job/wip%252F32886-add-tests-for-cook/1/testReport/junit/tests\\_schemas/test\\_cook/test\\_cook\\_ozwillo/](https://jenkins2.entrouvert.org/job/hobo-wip/job/wip%252F32886-add-tests-for-cook/1/testReport/junit/tests_schemas/test_cook/test_cook_ozwillo/)

En tout cas pour celle-ci ça me paraît parfait pour une CookError('unknown theme %s' % theme\_id) justement mais pas dans ce ticket, ici pour que ça fonctionne je dirai qu'il faut juste avoir une fixture qui crée un répertoire y met un faux thème et qui définit settings.THEMES\_DIRECTORY pour que hobo.theme.utils.get\_themes() fonctionne.

La seconde question (erreur avec tox, désolé d'avoir embrouillé le ticket) c'est de savoir si c'est envisageable de modifier jenkins pour prendre en compte les thèmes. Je dis ça parce qu'il faut que les thèmes ne soient pas installés (via make install) pour reproduire la trace jenkins ci-dessus.

Réponse dans mon dernier paragraphe, non le test doit être indépendant du paquet publik-base-theme.

#### #9 - 09 mai 2019 16:51 - Nicolas Roche

- Fichier 0001-cook-add-unit-tests-for-cook.py-32886.patch ajouté

- Fichier 0002-cook-add-tests-using-tenants-for-cook.py-32886.patch ajouté

- Statut changé de Information nécessaire à Solution proposée

voici donc deux patchs qui ajoutent des tests pour cook.py :

- tests/cook.py pour les tests unitaires
- tests\_schemas/cook.py : ajoute un scénario plausible côté serveur

#### #10 - 09 mai 2019 19:33 - Benjamin Dauvergne

Nicolas Roche a écrit :

voici donc deux patchs qui ajoutent des tests pour cook.py :

- tests/cook.py pour les tests unitaires
- tests\_schemas/cook.py : ajoute un scénario plausible côté serveur

Pour faire des répertoires temporaires tu as une fixture tmpdir[1] qui gère tout pour toi (mais ça renvoie des objets Path de pathlib il faut faire str()) dessus avant des les utiliser dans du code normal).

<sup>1</sup><https://docs.pytest.org/en/latest/tmpdir.html>

#### #11 - 09 mai 2019 19:59 - Benjamin Dauvergne

- test\_handle c'est un peu exagéré on ne descend pas aussi bas dans nos tests en général, enfin on va le garder
- Ça c'est assez étrange :

```
assert str(command.check_action.mock_calls) == expected1
```

Il me semble que tu peux directement instancier des objets call() et qu'ils sont comparables<sup>1</sup> tant que leurs arguments le sont. Ça devrait rendre tout ça plus lisible.

<sup>1</sup> Et oui

```
In [1]: from mock import call
```

```
In [2]: call('a', object) == call(u'a', object)
```

```
Out[2]: True
```

#### #12 - 10 mai 2019 11:03 - Nicolas Roche

- Fichier `.coverage` ajouté

- Fichier `0001-cook-add-unit-tests-for-cook.py-32886.patch` ajouté

- Fichier `0002-cook-add-tests-using-tenants-for-cook.py-32886.patch` ajouté

merci pour call, c'est beaucoup plus simple comme ça !

- fixture tmpdir
- comparaisons des appels via objets call

#### #13 - 10 mai 2019 11:03 - Nicolas Roche

- Fichier `.coverage` supprimé

#### #14 - 10 mai 2019 11:20 - Benjamin Dauvergne

C'est possible de pas avoir de `expected` et de mettre directement la valeur sur la ligne `assert` ? C'est plus lisible je trouve.

#### #15 - 10 mai 2019 15:33 - Nicolas Roche

- Fichier `0001-cook-add-unit-tests-for-cook.py-32886.patch` ajouté

- Fichier `0002-cook-add-tests-using-tenants-for-cook.py-32886.patch` ajouté

voilà

#### #16 - 10 mai 2019 15:43 - Benjamin Dauvergne

Nicolas Roche a écrit :

voilà

Ça ne doit pas être les bons patchs :

```
expected = [call('recipe_me.json')]
```

#### #17 - 10 mai 2019 18:59 - Nicolas Roche

- Fichier `0001-cook-add-unit-tests-for-cook.py-32886.patch` ajouté

- Fichier `0002-cook-add-tests-using-tenants-for-cook.py-32886.patch` ajouté

non en effet, désolé.

#### #18 - 17 mai 2019 19:11 - Nicolas Roche

- Sujet changé de tests sur `cook.py` à tests sur `cook.py` (côté serveur)

#### #19 - 21 mai 2019 15:24 - Benjamin Dauvergne

- receipe -> recipe
- plutôt qu'un mock d'open posé le fichier dans un tmpdir

```
with tmpdir / 'recipe.json' as path:
    with path.open() as fd:
        fd.write(json.dumps(recipe))
    command.run_cook(str(path))
```

je ne suis pas fan d'utiliser mock quand ce n'est pas nécessaire et qu'on peut tester le vrai code

- plus court que

```
value = Variable.objects.get(name='foo').value
label = Variable.objects.get(name='foo').label
assert value == 'bar'
assert label == 'foo'
```

on peut

```
variables = Variable.objects
...
var = variables.get(name='foo')
assert var.value == 'bar'
assert var.label == 'foo'
```

et

```
value = Variable.objects.get(name='foo').value
ordered_dump = json.dumps(json.loads(value), sort_keys=True)
assert ordered_dump == '{"key1": "bar", "key2": "baz}"'
```

plutôt

```
var = variables.get(name='foo')
assert var.json == '{"key1": "bar", "key2": "baz}"'
```

- je ne comprends pas cette remarque, la virer si ce n'est plus d'actualité

```
"""
code redondant ?
service_type=obj_type,
service_pk=obj.id,
"""
```

- je n'avais pas remarqué avant mais je ne vois pas le rapport avec ozwillo (faudrait renommer ça autrement)

#### #20 - 24 mai 2019 18:24 - Nicolas Roche

- Fichier 0001-cook-add-unit-tests-for-cook.py-32886.patch ajouté

merci benjamin, j'ai pris en compte tes remarques et j'ai mergé les 2 patches.

pour create\_site, je ne suis pas expert ORM mais d'après la doc :

<https://docs.djangoproject.com/fr/1.11/ref/models/querysets/#get-or-create>

```
variable, created = Variable.objects.get_or_create(name=variable_name,
service_type=obj_type,
service_pk=obj.id,
defaults={'label': label or variable_name})
```

fait un select sur le nom ET sur 2 paramètres supplémentaires,  
si ne trouve rien, alors crée cet enregistrement avec ces 2 paramètres.  
Dans les 2 cas on aura :

- service\_type=obj\_type,
- service\_pk=obj.id,

Voilà pourquoi je pense que les 2 lignes qui suivent sont redondantes :

```
variable.service_type = obj_type
variable.service_pk = obj.id
```

#### #21 - 24 mai 2019 19:35 - Benjamin Dauvergne

Je veux bien mais je ne vois pas de quels lignes tu parles.

#### #22 - 24 mai 2019 20:07 - Nicolas Roche

c'est dans hobo/environment/management/commands/cook.py::create\_site()  
et c'était à propos de ma remarque sur du code redondant :

```
for variable_name in variables.keys():
    label = variables[variable_name].get('label')
    variable, created = Variable.objects.get_or_create(name=variable_name,
service_type=obj_type,
service_pk=obj.id,
defaults={'label': label or variable_name})
    variable.service_type = obj_type # <-- ici
```

```
variable.service_pk = obj.id # <-- et là
```

### #23 - 24 mai 2019 20:48 - Benjamin Dauvergne

Nicolas Roche a écrit :

```
c'est dans hobo/environment/management/commands/cook.py::create_site()
et c'était à propos de ma remarque sur du code redondant :
[...]
```

Oui tout à fait c'est redondant :)

### #24 - 24 mai 2019 20:50 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

C'est ok pour moi, un petit coup de json\_pp -json\_opt pretty sur tests\_schemas/example\_env.json j'aime bien quand les fixtures sont lisibles.

### #25 - 27 mai 2019 19:38 - Nicolas Roche

- Fichier 0001-cook-add-unit-tests-for-cook.py-32886.patch ajouté

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

```
commit 2afc250a9796447bd02e408aa2bed94aac861945
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Mon May 27 19:35:36 2019 +0200
```

```
cook: add unit tests for cook.py (#32886)
```

### #26 - 27 mai 2019 19:43 - Nicolas Roche

- Lié à Bug #33457: code redondant ajouté

### #27 - 29 mai 2019 11:20 - Nicolas Roche

- Lié à Bug #33536: tests sur cook.py (correction) ajouté

### #28 - 31 mai 2019 09:15 - Frédéric Péters

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

## Fichiers

0001-cook-add-unit-tests-for-cook.py-32886.patch	15,5 ko	07 mai 2019	Nicolas Roche
0002-cook-add-tests-using-tenants-for-cook.py-32886.patch	4,7 ko	07 mai 2019	Nicolas Roche
0001-cook-add-unit-tests-for-cook.py-32886.patch	16,2 ko	09 mai 2019	Nicolas Roche
0002-cook-add-tests-using-tenants-for-cook.py-32886.patch	4,97 ko	09 mai 2019	Nicolas Roche
0001-cook-add-unit-tests-for-cook.py-32886.patch	16 ko	10 mai 2019	Nicolas Roche
0002-cook-add-tests-using-tenants-for-cook.py-32886.patch	4,97 ko	10 mai 2019	Nicolas Roche
0002-cook-add-tests-using-tenants-for-cook.py-32886.patch	4,97 ko	10 mai 2019	Nicolas Roche
0001-cook-add-unit-tests-for-cook.py-32886.patch	16 ko	10 mai 2019	Nicolas Roche
0002-cook-add-tests-using-tenants-for-cook.py-32886.patch	4,97 ko	10 mai 2019	Nicolas Roche
0001-cook-add-unit-tests-for-cook.py-32886.patch	15,3 ko	10 mai 2019	Nicolas Roche
0001-cook-add-unit-tests-for-cook.py-32886.patch	26,2 ko	24 mai 2019	Nicolas Roche
0001-cook-add-unit-tests-for-cook.py-32886.patch	30,2 ko	27 mai 2019	Nicolas Roche