

## Combo - Bug #10345

### toujours accéder à l'objet "user" via context['request'] (django 1.8)

19 mars 2016 17:02 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	19 mars 2016
<b>Priorité:</b>	Normal	<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		
<b>Description</b>			

#### Révisions associées

##### Révision 130a6c70 - 20 mars 2016 18:46 - Frédéric Péters

general: make sure context processors are applied in ajax rendering (#10345)

As Django 1.8 now supports several template engines the template context processors are only applied when the context is bound to an actual template; trick it here as a stopgap measure as we continue to support both Django 1.7 and 1.8.

#### Historique

##### #1 - 19 mars 2016 17:06 - Frédéric Péters

- Fichier 0001-general-always-get-user-object-from-request-10345.patch ajouté
- Statut changé de Nouveau à En cours
- Patch proposed changé de Non à Oui

##### #2 - 20 mars 2016 11:31 - Benjamin Dauvergne

C'est quoi le souci avec context.get('user') ? Il me semble que django.contrib.auth.context\_processors.auth fait toujours partie de la liste officielle des manipulateurs de contexte par défaut (<https://docs.djangoproject.com/fr/1.9/ref/settings/#template-context-processors>).

##### #3 - 20 mars 2016 11:47 - Frédéric Péters

Dans la pratique, je travaillais sur les cellules "factures" et

```
def get_items(self, context, past=False):
    """
    returns current or past items
    """
    if not self.is_remote():
        payed = not past
        return self.basketitem_set.filter(payment_date__isnull=payed,
                                         user=context.get('user'))
    if context.get('user'):
```

Et là il n'y avait pas de context['user'].

```
(Pdb) p context
[{'False': False, 'None': None, 'True': True}, {'synchronous': True, 'site_base': 'http://combo-test-theme.127.0.0.1.xip.io:8000', 'request': <WSGIRequest: GET '/ajax/cell/11/lingo_activeitems-1/'>, 'page': <Page: factures>}, {}]
```

Liste des context processors :

```
('django.core.context_processors.request', 'hobo.context_processors.theme_base', 'django.contrib.auth.context_processors.auth', 'django.template.context_processors.debug', 'django.template.context_processors.i18n', 'django.template.context_processors.media', 'django.template.context_processors.static', 'django.template.context_processors.tz', 'django.contrib.messages.context_processors.messages', 'combo.context_processors.template_vars')
```

Et le fond de l'histoire il est révélé dans

<http://stackoverflow.com/questions/34500977/django-upgrade-from-1-7-to-1-8-need-variable-from-contextprocessor-in-view> : « The upshot of this is that a RequestContext is not associated with any particular set of context processors until it is associated with an actual template. ».

#### #4 - 20 mars 2016 13:41 - Benjamin Dauvergne

Ouais, sacré sac de noeud. Néanmoins je pense que le problème ne se pose que dans la vue de rendu ajax qui justement n'a pas de template pour définir le contexte de résolution des manipulateurs de contextes; même si c'est très moche je propose de suivre la solution proposée dans le thread stackoverflow parce que sinon il va falloir se souvenir qu'on a pas accès à user (et aussi toutes les autres variables comme wcs\_url etc..) aussi dans les templates eux même quand une cellule en utilise un.

```
diff --git a/combo/public/views.py b/combo/public/views.py
index 3e68e8e..1f04ad1 100644
--- a/combo/public/views.py
+++ b/combo/public/views.py
@@ -28,6 +28,7 @@ from django.shortcuts import render, resolve_url
     from django.template import RequestContext
     from django.utils.translation import ugettext as _
     from django.forms.widgets import Media
+from django.template import loader

    try:
        from mellon.utils import get_idps
@@ -83,7 +84,11 @@ def ajax_page_cell(request, page_pk, cell_reference):
        # Cell can pass data through its own __dict__
        cell.modify_global_context(context, request)

-    return HttpResponse(cell.render(context), content_type='text/html')
+    # HACK: we need to bind to an existing template have the list of context processors set in the
+    # RequestContext
+    template = loader.get_template('combo/page_template.html')
+    with context.bind_template(template):
+        return HttpResponse(cell.render(context), content_type='text/html')

def extend_with_locked_placeholders_cells(cells, page, pages):
```

#### #5 - 20 mars 2016 14:59 - Frédéric Péters

- Fichier 0001-general-make-sure-context-processors-are-applied-in-.patch ajouté

En fait, ça doit être `bind_template(template.template)`, pour contourner des bouts de compatibilité assurés par Django; et de manière générale la méthode "render()" des cellules doit évoluer pour accepter un dictionnaire et la request en paramètre, pour avoir la même signature que les templates >= 1.8.

(de backends/django.py) :

```
def render(self, context=None, request=None):
    # A deprecation path is required here to cover the following usage:
    # >>> from django.template import Context
    # >>> from django.template.loader import get_template
    # >>> template = get_template('hello.html')
    # >>> template.render(Context({'name': 'world'}))
    # In Django 1.7 get_template() returned a django.template.Template.
    # In Django 1.8 it returns a django.template.backends.django.Template.
    # In Django 2.0 the isinstance checks should be removed. If passing a
    # Context or a RequestContext works by accident, it won't be an issue
    # per se, but it won't be officially supported either.
```

#### #6 - 20 mars 2016 18:35 - Benjamin Dauvergne

Ack.

#### #7 - 20 mars 2016 18:47 - Frédéric Péters

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

```
commit 130a6c70ec9a23f83b4a14f59bb5e08f5c575b97
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Sun Mar 20 14:53:58 2016 +0100
```

```
general: make sure context processors are applied in ajax rendering (#10345)
```

```
As Django 1.8 now supports several template engines the template context
processors are only applied when the context is bound to an actual template;
trick it here as a stopgap measure as we continue to support both Django 1.7
```

and 1.8.

## #8 - 28 juin 2016 20:07 - Frédéric Péters

- Statut *changé de Résolu* (à déployer) à *Fermé*

### Fichiers

---

0001-general-always-get-user-object-from-request-10345.patch	7,44 ko	19 mars 2016	Frédéric Péters
0001-general-make-sure-context-processors-are-applied-in-.patch	2,31 ko	20 mars 2016	Frédéric Péters