

## Combo - Development #17228

### plugin "nanterre" (saga)

28 juin 2017 12:08 - Thomas Noël

<b>Statut:</b>	Fermé	<b>Début:</b>	28 juin 2017
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Thomas Noël	<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 système de paiement à Nanterre est spécifique, il passe par des webservices SAGA qui se chargent de piloter TIPI. En gros, on doit envoyer à SAGA uniquement la liste des factures que l'utilisateur veut payer. SAGA se charge de nous dire l'URL TIPI à appeler.

### Historique

#### #1 - 28 juin 2017 12:09 - Thomas Noël

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

Voilà, c'est visible ici : <http://git.entrouvert.org/combo-plugin-nanterre.git/?h=wip>

#### #2 - 28 juin 2017 16:05 - Benjamin Dauvergne

- Assigné à mis à Thomas Noël

#### #3 - 28 juin 2017 16:09 - Benjamin Dauvergne

```
try:
    saga = requests.post(url, json=payload, remote_service=None, cache_duration=0, timeout=10).json()
except:
    messages.error(request, ERROR_MESSAGE)
    return HttpResponseRedirect(error_url)
```

Un peu violent il faudrait au moins logger l'erreur. Le seul cas où ça arrivera c'est si zoo est down, je pense qu'on voudrait être au courant.

#### #4 - 28 juin 2017 16:24 - Benjamin Dauvergne

Globalement je mettrai quelques logs en plus, du log.info() sur initiation réussie du paiement, retour synchrone/asynchrone réussi (on trouve un idop dans le payload).

Pour les erreurs de retour je loggerai simplement en warning.

```
RequestContext(request, {'request': request})
```

Il me semble que le but de RequestContext c'est déjà de mettre request dans le contexte (via les CONTEXT\_PROCESSOR).

Plutôt que de poser l'URL de retour dans un setting, est-ce qu'on ne pourrait pas la passer à la vue saga\_transaction:

```
diff --git a/combo_plugin_nanterre/views.py b/combo_plugin_nanterre/views.py
index eflb175..31dca6e 100644
--- a/combo_plugin_nanterre/views.py
+++ b/combo_plugin_nanterre/views.py
@@ -24,6 +24,7 @@ from django.core.urlresolvers import reverse
 from django.http import HttpResponseRedirect
 from django.template import RequestContext
 from django.utils.translation import ugettext_lazy as _
+from django.utils.http import urlencode
 from django.views.decorators.csrf import csrf_exempt

 from combo.utils import requests, get_templated_url
@@ -40,6 +41,10 @@ MESSAGE_BY_STATE = {
 @csrf_exempt
 @login_required
 def saga_transaction(request):
```

```

+ next_url = request.POST.get('next') or request.META.get('HTTP_REFERER')
+ # we accept only relative URLs, to prevent controlling the redirector
+ if not next_url.startswith('/') or next_url.startswith('///'):
+     next_url = '/'
+     num_factures = request.POST.getlist('num_factures')
+     email = request.POST.get('email') or request.user.email
+     error_url = request.POST.get('error_url') or '/'
@@ -51,6 +56,7 @@ def saga_transaction(request):
+     urlretour_asynchrone = urlretour_asynchrone[:-1]
+     if urlretour_synchrone.endswith('/'):
+         urlretour_synchrone = urlretour_synchrone[:-1]
+     urlretour_synchrone += '?' + urlencode({'next': next_url})

    payload = {
        'num_factures': num_factures,
@@ -93,21 +99,22 @@ def saga_retour(request):
    @csrf_exempt
    @login_required
    def saga_retour_synchrone(request):
-     url = getattr(settings, 'NANTERRE_PAYMENT_RESULT_PAGE', '/')
+     next_url = request.GET.get('next')
+     if not next_url.startswith('/') or next_url.startswith('///'):
+         next_url = '/'
    try:
        saga = saga_retour(request)
    except:
        messages.error(request, ERROR_MESSAGE)
-     return HttpResponseRedirect(url)
-
-     # add a result message and redirect
-     if isinstance(saga, dict) and saga.get('err') == 0 and saga.get('data', {}).get('etat'):
-         etat = saga['data']['etat']
-         if etat in MESSAGE_BY_STATE:
-             messages.add_message(request, *MESSAGE_BY_STATE[etat])
-         else:
-             messages.error(request, ERROR_MESSAGE)
-     return HttpResponseRedirect(url)
+ else:
+     # add a result message and redirect
+     if isinstance(saga, dict) and saga.get('err') == 0 and saga.get('data', {}).get('etat'):
+         etat = saga['data']['etat']
+         if etat in MESSAGE_BY_STATE:
+             messages.add_message(request, *MESSAGE_BY_STATE[etat])
+         else:
+             messages.error(request, ERROR_MESSAGE)
+     return HttpResponseRedirect(next_url)

@csrf_exempt
def saga_retour_asynchrone(request):

```

Bon sinon niveau PEP8 c'est pas ça, lignes trop longues, indentation à 8 caractères, nombre de sauts entre les fonctions, `ugettext_lazy` est inutilisé.

#### #5 - 28 juin 2017 16:41 - Benjamin Dauvergne

Vire ça pour la seule bonne façon sauf si on est pas vraiment sûr :)

```

idop = request.GET.get('idop')
if not idop:
    try:
        idop = json.loads(request.body).get('idop')
    except:
        pass

```

#### #7 - 28 juin 2017 18:01 - Thomas Noël

Benjamin Dauvergne a écrit :

Un peu violent il faudrait au moins logger l'erreur. Le seul cas où ça arrivera c'est si zoo est down, je pense qu'on voudrait être au courant.

En fait `combo.utils.requests` loggue déjà les erreurs.

#### #8 - 28 juin 2017 19:33 - Thomas Noël

J'ai mis à jour la branche <http://git.entrouvert.org/combo-plugin-nanterre.git/?h=wip>

avec les conseils de Benjamin :

- surtout, des logs, même quand tout marche (après tout, c'est la killerapp, le paiement)
- recherche du idop seulement dans la query string
- RequestContext(request, {'request': request}) : laissé comme ça, sinon ça marche pas, certainement un truc raté dans la compréhension ou dans la config des context processor
- pep8 is now my friend

et par ailleurs la possibilité de moduler le timeout dans le settings, si on voit que 30s sont nécessaires (oui, ça m'est arrivé).

#### #9 - 12 juillet 2017 19:36 - Frédéric Péters

```
return requests.post(url, json=payload,
                    cache_duration=0,
                    timeout=timeout).json()
```

cache\_duration est pris en compte uniquement pour les GET, inutile ici de le préciser ici.

```
# TIPI-WS refuse that an URL ends with a slash
if urlretour_asynchrone.endswith('/'):
    urlretour_asynchrone = urlretour_asynchrone[:-1]
if urlretour_synchrone.endswith('/'):
    urlretour_synchrone = urlretour_synchrone[:-1]
```

.rstrip('/') ?

error on creating transaction

un peu plus haut on a "failed to create transaction" qui est quand même mieux.

Dans la structure générale de saga\_transaction je pense que j'aurais plutôt préféré l'ensemble des cas d'erreur avant le "OK : redirect to payment system", genre :

```
if not isinstance(saga, dict):
    logger "mauvais format" et envoyer sur error_url
if saga.get('errors'):
    messages.error() de chacune
    logger les erreurs et envoyer sur error_url
if saga.get('err') != 0:
    logger "erreur inconnue ?" et envoyer sur error_url

# enfin bon
try:
    logger le succès de saga['data']['url']
    envoyer vers saga['data']['url']
except KeyError:
    logger "oops ça devrait pas arriver" et envoyer sur error_url
```

(en fait c'est beaucoup que je n'aime pas les isinstance, je pense, mais fais comme tu trouves le plus clair pour toi)

#### #10 - 13 juillet 2017 01:05 - Thomas Noël

Merci pour la relecture, d'accord avec tout, j'ai envoyé les modifs sur <http://git.entrouvert.org/combo-plugin-nanterre.git/?h=wip>

Si c'est OK, je fais de tout cela un seul commit sur "django code: urls and views".

#### #11 - 13 juillet 2017 07:36 - Frédéric Péters

Si c'est OK, je fais de tout cela un seul commit sur "django code: urls and views".

Ok en l'appelant plutôt "implement support for SAGA payment system".

#### #12 - 13 juillet 2017 11:06 - Thomas Noël

- Statut changé de En cours à Fermé

C'est fait ; merci.