

## Combo - Bug #13124

### lingo: moins dépendre d'une request

08 septembre 2016 14:18 - Thomas Noël

<b>Statut:</b>	Fermé	<b>Début:</b>	08 septembre 2016
<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

Dans lingo, beaucoup du code dépend d'un objet request (recherche du user meillon, signature de l'url pour les régies distantes) et ça va pas être facile pour [#13122](#), ie les choses à faire "hors requête"

#### Demandes liées:

Lié à Combo - Development #13122: lingo: systeme de notification en cas de fa...	Fermé	08 septembre 2016	09 mars 2018
Lié à Combo - Development #13132: utilisation de utils.requests avec un user ...	Rejeté	09 septembre 2016	
Lié à Combo - Development #13710: lingo: utiliser le système de signature pou...	Fermé	21 octobre 2016	

#### Révisions associées

##### Révision 005d0032 - 21 octobre 2016 15:03 - Thomas Noël

lingo: get name id from user object instead of session (#13124)

#### Historique

##### #1 - 08 septembre 2016 14:19 - Thomas Noël

- Lié à Development #13122: lingo: systeme de notification en cas de facture à payer ajouté

##### #2 - 09 septembre 2016 16:11 - Thomas Noël

- Lié à Development #13132: utilisation de utils.requests avec un user en dehors d'une requête ajouté

##### #3 - 21 octobre 2016 10:57 - Thomas Noël

- Fichier 0001-lingo-Regie-methods-no-longer-need-a-request-only-a-.patch ajouté

Voilà le patch, mais ça fait planter des tests dont je ne comprends même pas le code dans tests/test\_lingo\_payment.py

```
tests/test_lingo_payment.py::test_successfull_items_payment FAILED
tests/test_lingo_payment.py::test_payment_callback FAILED
```

Sur cette erreur :

```
===== FAILURES =====
=====
_____ test_successfull_items_payment _____
```

```
regie = <Regie: Test>, user = <User: admin>
```

```
def test_successfull_items_payment(regie, user):
    items = {'item1': {'amount': '10.5', 'source_url': '/item/1'},
            'item2': {'amount': '42', 'source_url': '/item/2'},
            'item3': {'amount': '100', 'source_url': '/item/3'},
            'item4': {'amount': '354', 'source_url': '/item/4'}}
    }
    b_items = []
    for subject, details in items.iteritems():
        b_item = BasketItem.objects.create(user=user, regie=regie,
                                          subject=subject, **details)
        b_items.append(b_item.pk)
    login()
    resp = client.post(reverse('lingo-pay'), {'item': b_items, 'regie': regie.pk})
```

```

assert resp.status_code == 302
location = resp.get('location')
assert 'dummy-payment' in location
parsed = urlparse.urlparse(location)
# get return_url and transaction id from location
qs = urlparse.parse_qs(parsed.query)
args = {'transaction_id': qs['transaction_id'][0], 'signed': True,
        'ok': True, 'reason': 'Paid'}
# make sure return url is the user return URL
assert urlparse.urlparse(qs['return_url'][0]).path.startswith(
    reverse('lingo-return', kwargs={'regie_pk': regie.id}))
# simulate successful return URL
resp = client.get(qs['return_url'][0], args)
assert resp.status_code == 302
assert urlparse.urlparse(resp.url).path == '/'
# simulate successful call to callback URL
> resp = client.get(reverse('lingo-callback', kwargs={'regie_pk': regie.id}), args)

```

```
tests/test_lingo_payment.py:129:
```

```

-----
./publik/venv/local/lib/python2.7/site-packages/django/test/client.py:500: in get
**extra)
./publik/venv/local/lib/python2.7/site-packages/django/test/client.py:303: in get
return self.generic('GET', path, secure=secure, **r)
./publik/venv/local/lib/python2.7/site-packages/django/test/client.py:379: in generic
return self.request(**r)
./publik/venv/local/lib/python2.7/site-packages/django/test/client.py:466: in request
six.reraise(*exc_info)
./publik/venv/local/lib/python2.7/site-packages/django/core/handlers/base.py:132: in get_response
response = wrapped_callback(request, *callback_args, **callback_kwargs)
./publik/venv/local/lib/python2.7/site-packages/django/views/generic/base.py:71: in view
return self.dispatch(request, *args, **kwargs)
./publik/venv/local/lib/python2.7/site-packages/django/views/decorators/csrf.py:58: in wrapped_view
return view_func(*args, **kwargs)
combo/apps/lingo/views.py:450: in dispatch
return super(CallbackView, self).dispatch(*args, **kwargs)
./publik/venv/local/lib/python2.7/site-packages/django/views/generic/base.py:89: in dispatch
return handler(request, *args, **kwargs)
combo/apps/lingo/views.py:443: in get
return self.handle_callback(request, request.environ['QUERY_STRING'], **kwargs)
combo/apps/lingo/views.py:432: in handle_callback
item.notify_payment()
combo/apps/lingo/models.py:192: in notify_payment
self.notify('paid')
combo/apps/lingo/models.py:188: in notify
data=json.dumps(message), headers=headers, timeout=3)
./publik/venv/local/lib/python2.7/site-packages/requests/sessions.py:522: in post
return self.request('POST', url, data=data, json=json, **kwargs)
combo/utils.py:162: in request
response = super(Requests, self).request(method, url, **kwargs)
./publik/venv/local/lib/python2.7/site-packages/requests/sessions.py:461: in request
prep = self.prepare_request(req)
./publik/venv/local/lib/python2.7/site-packages/requests/sessions.py:394: in prepare_request
hooks=merge_hooks(request.hooks, self.hooks),
./publik/venv/local/lib/python2.7/site-packages/requests/models.py:294: in prepare
self.prepare_url(url, params)

```

```
self = <PreparedRequest [POST]>, url = '/item/2jump/trigger/paid', params = OrderedDict()
```

```

def prepare_url(self, url, params):
    """Prepares the given HTTP URL."""
    #: Accept objects that have string representations.
    #: We're unable to blindly call unicode/str functions
    #: as this will include the bytestring indicator (b'')
    #: on python 3.x.
    #: https://github.com/kennethreitz/requests/pull/2238
    if isinstance(url, bytes):
        url = url.decode('utf8')
    else:
        url = unicode(url) if is_py2 else str(url)

    # Don't do any URL preparation for non-HTTP schemes like `mailto`,

```

```
# `data` etc to work around exceptions from `url_parse`, which
# handles RFC 3986 only.
if ':' in url and not url.lower().startswith('http'):
    self.url = url
    return
```

```
# Support for unicode domain names and paths.
try:
    scheme, auth, host, port, path, query, fragment = parse_url(url)
except LocationParseError as e:
    raise InvalidURL(*e.args)
```

```
if not scheme:
    error = ("Invalid URL {0!r}: No schema supplied. Perhaps you meant http://{0}?")
    error = error.format(to_native_string(url, 'utf8'))
```

```
> raise MissingSchema(error)
E MissingSchema: Invalid URL '/item/2jump/trigger/paid': No schema supplied. Perhaps you meant http://item/2jump/trigger/paid?
```

```
../publik/venv/local/lib/python2.7/site-packages/requests/models.py:354: MissingSchema
```

```
----- Captured stderr call -----
```

```
No handlers could be found for logger "combo.apps.lingo.views"
```

```
WARNING:root:service not found in settings.KNOWN_SERVICES for /item/2jump/trigger/paid
```

Et bon, en fait, je ne comprends pas les « 'source\_url': '/item/1' »

#### #4 - 21 octobre 2016 11:00 - Frédéric Péters

- *Projet changé de Combo à Lingo*

#### #5 - 21 octobre 2016 11:08 - Frédéric Péters

Les source\_url sont ici bidons et le résultat du code actuel c'est qu'il n'y a pas de notification envoyée, parce qu'aucun service n'est trouvé qui corresponde. Avec le code modifié c'est différent, tout le temps il est essayé de faire un POST; peut-être qu'il faudrait une nouvelle fonction, pour faire if not source\_from\_known\_orig(item.source\_url): return.

#### #6 - 21 octobre 2016 11:20 - Thomas Noël

- *Projet changé de Lingo à Combo*

Oui mais alors ce que je ne comprends pas : dans le code actuel, quand "aucun service n'est trouvé qui corresponde" notify devrait crasher (raise RuntimeError('failed to find data for server')), non ?

#### #7 - 21 octobre 2016 11:21 - Thomas Noël

ok, vu :

```
try:
    item.notify_payment()
except RuntimeError:
    # ignore errors, it should be retried later on if it fails
    pass
```

#### #8 - 21 octobre 2016 12:00 - Thomas Noël

- *Fichier 0001-lingo-Regie-methods-no-longer-need-a-request-only-a-patch ajouté*

- *Patch proposed changé de Non à Oui*

Patch où les tests sont ok ; j'ai préféré adapter les tests pour les rendre plus réalistes (et au passage je regarde vite si le callback lancera bien un notify).

#### #9 - 21 octobre 2016 12:40 - Frédéric Péters

J'aurais bien vu trois patches :

- lingo: use general signature scheme to get invoices
- lingo: get name id from user object instead of session [qui correspond au titre du ticket]
- lingo: use general signature scheme to send notifications

Mais démêler les deux premiers va être trop galère :/ Du coup tant pis et entre les deux messages de commit je garderais le second. Facile à sortir par contre, il me semble, c'est le troisième patch.

À côté de ça, comme on en est à changer les signatures de fonction, je suggérerais d'en profiter pour avoir s/item/invoice/.

```
return self.basketitem_set.filter(payment_date__isnull=not history, user=user)
```

Je trouve plus lisible de faire bool(not history). (et je comprends qu'on utilise "history" parce que ça correspond au endpoint mais c'est quand même plutôt pas terrible comme nom, tant pis).

#### #10 - 21 octobre 2016 14:12 - Thomas Noël

- Fichier 0001-lingo-get-name-id-from-user-object-instead-of-sessio.patch ajouté

Voilà le patch «get name id from user object», avec des s/item/invoice/ là où j'ai trouvé que ça avait du sens. Il reste des "item" dans le code parce qu'encore du mélange avec les BasketItems, et que je préférerai un patch séparé pour finir ce nettoyage.

#### #11 - 21 octobre 2016 14:12 - Thomas Noël

- Projet changé de Combo à Lingo

#### #12 - 21 octobre 2016 14:14 - Thomas Noël

- Lié à Development #13710: lingo: utiliser le système de signature pour les notifications de paiement ajouté

#### #13 - 21 octobre 2016 14:51 - Frédéric Péters

ok.

#### #14 - 21 octobre 2016 15:04 - Thomas Noël

- Statut changé de Nouveau à Résolu (à déployer)

```
commit 005d003292560289ff0e47c02a86656ea52b9d6a
Author: Thomas NOEL <tnoel@entrouvert.com>
Date: Fri Oct 21 14:06:20 2016 +0200
```

```
lingo: get name id from user object instead of session (#13124)
```

#### #15 - 26 octobre 2016 15:41 - Serghei Mihai (congé, retour 15/05)

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

Erreur sur la dev:

```
NameError: global name 'boot' is not defined
(2 additional frame(s) were not displayed)
...
File "combo/apps/lingo/models.py", line 454, in render
    return super(Items, self).render(context)
File "combo/data/models.py", line 437, in render
    context.update(self.get_cell_extra_context(context))
File "combo/apps/lingo/models.py", line 445, in get_cell_extra_context
    items = self.get_invoices(user=context['user'])
File "combo/apps/lingo/models.py", line 479, in get_invoices
    items.extend(r.get_invoices(user))
File "combo/apps/lingo/models.py", line 125, in get_invoices
    return self.basketitem_set.filter(payment_date__isnull=boot(not history), user=user)
```

#### #16 - 26 octobre 2016 15:51 - Frédéric Péters

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

[#13757](#).

#### #17 - 23 décembre 2018 15:23 - Frédéric Péters

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

### Fichiers

0001-lingo-Regie-methods-no-longer-need-a-request-only-a.patch	14,3 ko	21 octobre 2016	Thomas Noël
0001-lingo-Regie-methods-no-longer-need-a-request-only-a.patch	17,3 ko	21 octobre 2016	Thomas Noël
0001-lingo-get-name-id-from-user-object-instead-of-sessio.patch	13,5 ko	21 octobre 2016	Thomas Noël