

Combo - Development #25725

retourner l'URL de paiement d'un montant, après son ajout dans le panier

20 août 2018 18:16 - Serghei Mihai

Statut:	Fermé	Début:	20 août 2018
Priorité:	Normal	Echéance:	19 septembre 2018
Assigné à:	Serghei Mihai	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Afin de pouvoir payer le montant posé par une demande wcs, sans le cumuler avec d'autres montants, posés par d'autres demandes.			
J'imagine cette url utilisée dans l'action "redirection vers une url" au lieu d'un message invitant l'utilisateur à se rendre vers la page contenant le panier.			
Demandes liées:			
Lié à Combo - Development #25552: Pouvoir renvoyer l'utilisateur directement vers ...		Fermé	31 juillet 2018

Révisions associées

Révision 13bbfa8b - 04 octobre 2018 11:51 - Serghei Mihai

lingo: refactor payment view (#25725)

Révision a7076e6e - 04 octobre 2018 11:51 - Serghei Mihai

lingo: handle single basket item payment (#25725)

Historique

#1 - 17 septembre 2018 10:27 - Serghei Mihai

- Echéance mis à 19 septembre 2018

- Assigné à mis à Serghei Mihai

#2 - 20 septembre 2018 16:03 - Serghei Mihai

- Fichier 0001-lingo-refactor-payment-view-25725.patch ajouté

- Fichier 0002-lingo-handle-single-basket-item-payment-25725.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Un refactoring de la vue de paiement d'abord, et puis renvoi de l'url de paiement au webservice appelant.

#3 - 21 septembre 2018 15:59 - Frédéric Péters

(patch 0001 plutôt impossible à relire avec du code qui se déplace dans tous les sens)

```
def handle_payment(self, request, regie, items, remote_items=[], next_url='/', email='')
```

Le `remote_items=[]` va gentiment nous être signalé par pylint comme une construction dangereuse. Comme en plus il n'est pas fait profit d'avoir des valeurs par défaut aux paramètres, je dirais de simplement retirer les valeurs par défaut. (au moins pour `remote_items`).

```
url(r'^lingo/regie/(?P<regie_id>[\w,-]+)/item/(?P<item_id>[\w,-]+)/pay$',  
BasketItemPayView.as_view(), name='basket-item-pay-view'),
```

Je ne pense pas utile d'inclure la régie ici (on peut l'obtenir via l'item). L'`item_id` pourrait être limité à des chiffres.

```
user = request.user if request.user.is_authenticated() else None  
if user is None:
```

Plutôt simplement `if request.user and request.user.is_authenticated():`.

On autorise ici un utilisateur loggué à payer un élément de panier d'une autre personne, est-ce volontaire ?

#4 - 21 septembre 2018 16:03 - Frédéric Péters

À défaut d'une solution plus intelligente, il faudrait refuser l'appel à cette vue si la régie définit `extra_fees_ws_url`. (parce que ça permettrait de payer individuellement les différentes pièces et de laisser genre les frais postaux de côté).

#5 - 22 septembre 2018 14:43 - Frédéric Péters

- Lié à [Development #25552](#): Pouvoir renvoyer l'usager directement vers l'historique d'une demande (`form_url`) lorsque l'usager paye en ligne dans le cadre d'une démarche ajouté

#6 - 24 septembre 2018 18:23 - Serghei Mihai

- Fichier `0002-lingo-handle-single-basket-item-payment-25725.patch` ajouté

- Fichier `0001-lingo-refactor-payment-view-25725.patch` ajouté

Premier patch revu pour être plus lisible.

Les autres remarques prises en compte aussi.

#7 - 24 septembre 2018 18:26 - Serghei Mihai

- Fichier `0002-lingo-handle-single-basket-item-payment-25725.patch` supprimé

#8 - 24 septembre 2018 18:26 - Serghei Mihai

- Fichier `0001-lingo-refactor-payment-view-25725.patch` supprimé

#9 - 24 septembre 2018 18:26 - Serghei Mihai

- Fichier `0002-lingo-handle-single-basket-item-payment-25725.patch` ajouté

- Fichier `0001-lingo-refactor-payment-view-25725.patch` ajouté

Les bons patches.

#10 - 24 septembre 2018 22:28 - Frédéric Péters

Une raison pour ne pas placer le `@atomic` sur le `handle_payment` ?

Message pas clair de ma part plus haut, dans `class BasketItemPayView`, je voulais dire que je ne voyais pas le sens à déporter la condition dans une affectation `user = ... if .. or ...`, la condition, en elle-même, c'était mieux de la garder pour faire la gestion de l'erreur au plus tôt, que le niveau d'indentation de base de la méthode soit le cas passant (surtout qu'ici il y a un `mix`). Tant qu'à gérer ces situations, soyons francs via un `HttpResponseForbidden`, plutôt que redirection + message; bref :

```
if not (request.user and request.user.is_authenticated()):
    return HttpResponseForbidden('No item payment for anonymous users')
...
if regie.extra_fees_ws_url:
    return HttpResponseForbidden('No targeted payment as extra fees are set')
...
```

#11 - 25 septembre 2018 21:15 - Serghei Mihai

- Fichier `0002-lingo-handle-single-basket-item-payment-25725.patch` ajouté

- Fichier `0001-lingo-refactor-payment-view-25725.patch` ajouté

Frédéric Péters a écrit :

Une raison pour ne pas placer le `@atomic` sur le `handle_payment` ?

Non, tu as raison.

Message pas clair de ma part plus haut, dans `class BasketItemPayView`, je voulais dire que je ne voyais pas le sens à déporter la condition dans une affectation `user = ... if .. or ...`, la condition, en elle-même, c'était mieux de la garder pour faire la gestion de l'erreur au plus tôt, que le niveau d'indentation de base de la méthode soit le cas passant (surtout qu'ici il y a un `mix`). Tant qu'à gérer ces situations, soyons francs via un `HttpResponseForbidden`, plutôt que redirection + message; bref :

Ok.

#12 - 26 septembre 2018 09:00 - Frédéric Péters

Il me semble que #25552 serait traité ici, pour être sûr tu pourrais allonger ton test pour vérifier le bon suivi des redirections ? (uniquement si c'est supposé fonctionner, sinon ça sera patch dans l'autre ticket).

#13 - 26 septembre 2018 10:04 - Serghei Mihai

Yep, c'est à traiter ici.

En plus c'est le test que je fais en local pour m'assurer que tout fonctionne.

Je rajoute des tests.

#14 - 26 septembre 2018 10:51 - Serghei Mihai

- Fichier 0001-lingo-handle-single-basket-item-payment-25725.patch ajouté

Dernier patch à jour avec le test de la redirection.

#15 - 26 septembre 2018 11:10 - Frédéric Péters

J'ai du mal à suivre le test, j'ai l'impression que comme avant ça crée la redirection vers dummy-payment.

```
assert resp.location.startswith('http://dummy-payment.demo.entrouvert.com/')
qs = urlparse.parse_qs(urlparse.urlparse(resp.location).query)
assert qs['amount'] == ['12.00']
...
resp = app.get(qs['return_url'][0], params=data)
```

Ça teste qu'on envoie "return_url" à dummy-payment, avec la bonne valeur, mais il me semble qu'on ne vérifie rien du retour. (si je regarde le code ça doit passer par un lingo_next_url dans la session ?).

#16 - 26 septembre 2018 11:18 - Serghei Mihai

Frédéric Péters a écrit :

J'ai du mal à suivre le test, j'ai l'impression que comme avant ça crée la redirection vers dummy-payment.

```
resp = app.get(qs['return_url'][0], params=data)
```

Simule le retour de l'utilisateur de la plateforme de paiement à travers le bouton "Retour au commerce". Ça renvoie vers Combo, qui redirige vers la next_url, stockée en session lors du lancement du paiement

#17 - 26 septembre 2018 11:29 - Frédéric Péters

Simule le retour de l'utilisateur de la plateforme de paiement à travers le bouton "Retour au commerce".

Je reste pas clair là-dessus, pour moi ça ne teste pas ce qu'on veut, qui est qu'après le paiement on arrive sur /lingo/return/... et qu'il doit y avoir traitement et redirection. Peut-être plus clairement : je me serais attendu à avoir /lingo/return/... dans ton test.

#18 - 26 septembre 2018 17:13 - Serghei Mihai

qs['return_url'][0] contient l'url de retour vers combo: /lingo/return/<regie>/.

Et à l'appel vers return_url (qui est fait par le backend de paiement), une redirection vers la next_url se fait.

Ou ton idée est d'appeler l'url de retour dans le style ?

```
app.get(reverse('lingo-return', kwargs={'regie_pk': ...}), params=data)
```

#19 - 26 septembre 2018 17:29 - Frédéric Péters

Le parcours, il est combo → payment → combo → next_url. Et ton test, il fait juste combo → payment, est-ce que j'ai donné à payment le nécessaire pour que derrière il y ait au final redirection à next_url, mais pas est-ce que le parcours → combo → next_url qui devrait suivre se joue vraiment correctement.

#20 - 27 septembre 2018 09:33 - Serghei Mihai

C'est ce que je fais:

- combo: lancement du paiement de l'élément du panier avec une next_url:

```
# set next_url to check if redirection is done after payment
resp = app.get(payment_url, params={'next_url': 'http://example.net/form/id/'})
```

- payment: vérification que la redirection est faite vers le backend

```
assert resp.location.startswith('http://dummy-payment.demo.entrouvert.com/')
```

- backend de payment -> combo:

```
resp = app.get(qs['return_url'][0], params=data)
```

Il manque ici la vérification que l'item est bien payé. Je vais le rajouter.

- vérification que combo fait la redirection vers next_url:

```
assert resp.location == 'http://example.net/form/id/'
```

#21 - 27 septembre 2018 12:33 - Frédéric Péters

```
resp = app.get(payment_url, params={'next_url': 'http://example.net/form/id/'})
assert resp.location.startswith('http://dummy-payment.demo.entrouvert.com/')
```

On est au moment où la redirection vers le système de paiement a lieu.

```
qs = urlparse.parse_qs(urlparse.urlparse(resp.location).query)
```

On tape dans qs la query string de cette redirection.

```
resp = app.get(qs['return_url'][0], params=data)
```

Et là en fait le return_url tapé ici, c'est l'adresse /lingo/return/XXX/, ce que je ne captais pas.

Je pense critique de taper des commentaires pour noter les différentes étapes, parce que même en essayant d'y être attentif c'est super chaud à suivre, et quand il s'agira de comprendre à nouveau ça dans trois mois, ça sera la même misère.

#22 - 28 septembre 2018 11:59 - Serghei Mihai

Patch à jour avec des commentaires et check que l'item est bien marqué comme payé.

#23 - 02 octobre 2018 16:14 - Frédéric Péters

Manque le patch.

#24 - 02 octobre 2018 16:19 - Serghei Mihai

- Fichier 0001-lingo-handle-single-basket-item-payment-25725.patch ajouté

Merci.

#25 - 02 octobre 2018 17:10 - Frédéric Péters

Et là en fait le return_url tapé ici, c'est l'adresse /lingo/return/XXX/, ce que je ne captais pas.

Et les deux lignes de commentaires ajoutés n'aident en rien. Je suis peut-être super con à lire ce test mais pour comprendre c'est plus facile pour moi d'avoir des commentaires qui décrivent les actions (sans empêcher les commentaires qui décrivent ce qui est testé), ex:

```
# simulate payment service redirecting the user to /lingo/return/... (eopayment
# dummy module put that URL in return_url query string parameter).
resp = app.get(...)
# check payment is marked as done
assert BasketItem.objects.filter(regie=regie, amount=amount, payment_date__isnull=False).exists()
# check user is redirected to the URL that was specified at the beginning
assert resp.location == 'http://example.net/form/id/'
```

#26 - 03 octobre 2018 10:08 - Serghei Mihai

- Fichier 0001-lingo-handle-single-basket-item-payment-25725.patch ajouté

Certainement c'est trop évident pour moi, mais pas clair à la lecture pour les autres.
Je rajoute plus de commentaires.

#27 - 03 octobre 2018 10:56 - Frédéric Péters

1. simulate successfull payment response from dummy backend

successful.

1. simulate payment backend redirecting to /ling/return/, which is passed by

lingo.

Après ça devient intéressant que ça soit désormais quelqu'un d'autre que moi qui relise pour voir si le déroulé des tests lui est compréhensible. (parce que moi je trouve toujours peu claire l'affaire, qui suppose que le lecteur sait que ce return_url se trouve dans la redirection parce que le backend dummy d'eopayment l'y met, ce qui n'est pas noté, et le lecteur de se perdre parce que tu écris "passed by lingo" et qu'on ne trouve pas de return_url dans le code, etc. c'est contre ça que je mettais ma parenthèse "(eopayment dummy module put that URL in return_url query string parameter)".)

(bref je passe la main)

#28 - 03 octobre 2018 11:08 - Serghei Mihai

- Fichier 0001-lingo-handle-single-basket-item-payment-25725.patch ajouté

Ok, je reprend intégralement ton commentaire.

#29 - 03 octobre 2018 22:31 - Benjamin Dauvergne

- ```
payment_url = reverse('basket-item-pay-view', kwargs={'item_id': item.id})
response.write(json.dumps({'result': 'success', 'id': str(item.id),
 'payment_url': request.build_absolute_uri(payment_url)}))
```

À ce stade c'est juste pour faire chier mais il existe `django.http.response.JsonResponse` déjà en Django 1.8

- ```
C'est de la prospective à ce stade mais c'est possible à un moment que '/' ne soit pas la racine du projet (si un jour on déploie sur des sous-chemins), il vaudrait mieux utiliser reverse(combo.public.views.page).
```

Sur les tests c'est plus des questions de style:

- ```
page.save()
cell = LingoBasketCell(page=page, placeholder='content', order=0)
cell.save()
```

`.objects.create` te ferait gagner une ligne (c'est discutable avec beaucoup d'arguments mais c'est plus clair je trouve)

- ```
'url': 'http://example.com' }
```

c'est PEP8 mais moi ça me fait mal aux yeux, je préfère que tout soit aligné ou tout sur une ligne mais pas le mélange

- ```
Toutes les réponses de webtest on un attribut .json, pas besoin de faire json.loads
```

## #30 - 03 octobre 2018 23:35 - Serghei Mihai

- Fichier 0001-lingo-handle-single-basket-item-payment-25725.patch ajouté

Benjamin Dauvergne a écrit :

- [...] À ce stade c'est juste pour faire chier mais il existe `django.http.response.JsonResponse` déjà en Django 1.8

Ok.

- [...] C'est de la prospective à ce stade mais c'est possible à un moment que '/' ne soit pas la racine du projet (si un jour on déploie sur des sous-chemins), il vaudrait mieux utiliser `reverse(combo.public.views.page)`.

Je préfère ne pas traiter ça ici.

Sur les tests c'est plus des questions de style:

- [...] `.objects.create` te ferait gagner une ligne (c'est discutable avec beaucoup d'arguments mais c'est plus clair je trouve)

Moi aussi, mais je suis le style des tests précédents.

- [...] c'est PEP8 mais moi ça me fait mal aux yeux, je préfère que tout soit aligné ou tout sur une ligne mais pas le mélange

Ok.

- [...] Toutes les réponses de webtest on un attribut `.json`, pas besoin de faire `json.loads`

Ok.

#### #31 - 04 octobre 2018 11:37 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Ack, mais c'est moche cette façon d'indenter les dicos et de garder le style pourri des tests précédents. Un dico ça doit ressembler à ça:

```
d = {
 "x": 1,
 "y": 2,
}
```

et je serai pas contre suivre la règle de l'inverse christmas tree du kernel + ordre alphanumérique:

```
d = {
 "longlonglong": 1,
 "longlong": 2,
 "c": 3,
}
```

c'est tout de même plus lisible.

#### #32 - 04 octobre 2018 11:55 - Serghei Mihai

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

Allez hop, on parler des goûts plus tard:

```
commit a7076e6e1850639d96b1dff5f10c11c042235459 (origin/master)
Author: Serghei Mihai <smihai@entrouvert.com>
Date: Wed Sep 19 17:46:16 2018 +0200
```

```
lingo: handle single basket item payment (#25725)
```

```
commit 13bbfa8bf1fde0f8e557d9c978c6b5648020417b
Author: Serghei Mihai <smihai@entrouvert.com>
Date: Mon Sep 24 17:56:36 2018 +0200
```

```
lingo: refactor payment view (#25725)
```

## Fichiers

---

|                                                          |         |                   |               |
|----------------------------------------------------------|---------|-------------------|---------------|
| 0001-lingo-refactor-payment-view-25725.patch             | 7,05 ko | 20 septembre 2018 | Serghei Mihai |
| 0002-lingo-handle-single-basket-item-payment-25725.patch | 5,27 ko | 20 septembre 2018 | Serghei Mihai |
| 0002-lingo-handle-single-basket-item-payment-25725.patch | 5,46 ko | 24 septembre 2018 | Serghei Mihai |
| 0001-lingo-refactor-payment-view-25725.patch             | 6,84 ko | 24 septembre 2018 | Serghei Mihai |
| 0002-lingo-handle-single-basket-item-payment-25725.patch | 5,74 ko | 25 septembre 2018 | Serghei Mihai |
| 0001-lingo-refactor-payment-view-25725.patch             | 6,83 ko | 25 septembre 2018 | Serghei Mihai |
| 0001-lingo-handle-single-basket-item-payment-25725.patch | 6,08 ko | 26 septembre 2018 | Serghei Mihai |
| 0001-lingo-handle-single-basket-item-payment-25725.patch | 6,41 ko | 02 octobre 2018   | Serghei Mihai |
| 0001-lingo-handle-single-basket-item-payment-25725.patch | 6,66 ko | 03 octobre 2018   | Serghei Mihai |
| 0001-lingo-handle-single-basket-item-payment-25725.patch | 6,7 ko  | 03 octobre 2018   | Serghei Mihai |
| 0001-lingo-handle-single-basket-item-payment-25725.patch | 7,25 ko | 03 octobre 2018   | Serghei Mihai |