

## Combo - Bug #11063

### redirection vers url en session

26 mai 2016 14:09 - Frédéric Péters

<b>Statut:</b> Nouveau	<b>Début:</b> 26 mai 2016
<b>Priorité:</b> Normal	<b>Echéance:</b>
<b>Assigné à:</b>	<b>% réalisé:</b> 0%
<b>Catégorie:</b>	<b>Temps estimé:</b> 0:00 heure
<b>Version cible:</b>	<b>Planning:</b> Non
<b>Patch proposed:</b> Non	

**Description**

```
if next_url:
    # store the next url in session in order to be able to redirect to
    # it if payment is canceled
    request.session.setdefault('lingo_next_url',
                               {})[transaction.order_id] = request.build_absolute_uri(next_url)
```

Mais au retour du service de paiement,

```
if request.session.get('lingo_next_url'):
    redirect_url = request.session['lingo_next_url'].get(transaction.order_id)
    if redirect_url:
        return HttpResponseRedirect(redirect_url)
```

On ne fait pas gaffe de savoir si c'est parce qu'il y a eu annulation et du coup on pourrait très bien rester sur une redirection vers la page contenant le panier, alors que celui-ci est vide et qu'on a du code supposé détecter ça et envoyer vers la page d'accueil dans cette situation.

**Demandes liées:**

Dupliqué par Combo - Bug #13600: ne pas rediriger vers lingo_next_url en cas ...	<b>Rejeté</b>	<b>14 octobre 2016</b>
--	---------------	------------------------

### Historique

#### #1 - 26 juillet 2016 22:43 - Frédéric Péters

- Description mis à jour

#### #2 - 11 février 2017 17:51 - Frédéric Péters

- Dupliqué par Bug #13600: ne pas rediriger vers lingo\_next\_url en cas de succès ajouté

#### #3 - 03 novembre 2020 13:02 - Benjamin Dauvergne

Je ne comprends pas cette demande, notamment le cas annulation, de quel annulation parle-t-on de l'Item ou du paiement coté fournisseur de paiement ?

Aussi c'est bien de faire un topo de ce qui se passe au niveau des retours:

On a effectivement du code qui renvoie soit vers le panier soit vers l'accueil dans ReturnView mais uniquement dans le cas où le retour n'a pas permis d'identifier une transaction dans tous les autres cas on va sur PaymentStatusView qui va :

- renvoyer vers lingo\_next\_url qui est :
  - depuis le panier, c'est l'url de la page où il est affiché (via paramètre next\_url de PayView)
  - depuis le paiement d'une facture, pour ActiveItems et SelfInvoice ça va être la page de ces cellules,
  - depuis un mail de notification, l'argument page étant vide, le champ caché next\_url dans item.html sera vide, et donc ce sera la page d'accueil
- ou si lingo\_next\_url est vide :
  - renvoyer vers l'URL source commune à tous les items payés si ils ont tous la même,
  - ou bien si pas d'URL commune unique, vers la page de la première cellule de panier trouvée.

#### #4 - 03 novembre 2020 13:23 - Frédéric Péters

Je ne comprends pas cette demande, notamment le cas annulation, de quel annulation parle-t-on de l'Item ou du paiement coté fournisseur de paiement ?

Le commentaire "if payment is canceled", il vient de [#9854](#), je ne vais pas chercher ce qu'il veut dire. Mais d'accord qu'il n'est pas clair.

De manière plus générale ce ticket date d'il y a plus de quatre ans (et celui doublon [#13600](#) pareil); visiblement j'étais à l'époque au moins deux fois dans ce code et à sérieusement m'interroger dessus, avec zéro retour.

À noter aussi que [#36876](#) a modifié des bouts en rapport

```
    # store the next url in session in order to be able to redirect to
    # it if payment is canceled
-   request.session.setdefault('lingo_next_url',
-                               {})[order_id] = request.build_absolute_uri(next_url)
+   if next_url:
+       request.session.setdefault('lingo_next_url',
+                                   {})[str(transaction.pk)] = request.build_absolute_uri(next_url)
```

et peut-être surtout celui-ci, qui remplace/déplace une utilisation de `lingo_next_url`,

```
-   if transaction and request.session.get('lingo_next_url'):
-       redirect_url = request.session['lingo_next_url'].get(transaction.order_id)
-       if redirect_url:
-           return HttpResponseRedirect(redirect_url)
+   if transaction:
+       return HttpResponseRedirect(get_payment_status_view(transaction.pk))
```

peut-être que tout ça avait moins de sens que maintenant il y a quatre ans quand je m'interrogeais sur le code et des situations qui pouvaient peut-être arriver, qui peut-être n'arrivent plus.

#### #5 - 03 novembre 2020 14:06 - Serghei Mihai

Frédéric Péters a écrit :

Le commentaire "if payment is canceled", il vient de [#9854](#), je ne vais pas chercher ce qu'il veut dire. Mais d'accord qu'il n'est pas clair.

En effet le commentaire est merdique. `lingo_next_url` a été introduit pour pouvoir rediriger un utilisateur vers la demande dans wcs qui a ajouté le montant dans le panier et a lancé le paiement, et ceci que le paiement réussisse ou échoue.

Donc pour compléter le topo de Benjamin:

- renvoyer vers `lingo_next_url` qui est :
  - depuis le panier, c'est l'url de la page où il est affiché (via paramètre `next_url` de `PayView`)
  - depuis le paiement d'une facture, pour `ActiveItems` et `SelfInvoice` ça va être la page de ces cellules,
  - depuis un mail de notification, l'argument `page` étant vide, le champ caché `next_url` dans `item.html` sera vide, et donc ce sera la page d'accueil
- l'url de la demande qui a ajouté le montant dans le panier et a lancé le paiement.

Mais le code a bougé depuis et dans `BasketItemPayView` on renvoie déjà vers la source de l'item:

```
if not next_url:
    next_url = item.source_url
```

Je pense qu'on peut virer ce `lingo_next_url`.

#### #6 - 03 novembre 2020 16:48 - Benjamin Dauvergne

De manière plus générale ce ticket date d'il y a plus de quatre ans (et celui doublon [#13600](#) pareil); visiblement j'étais à l'époque au moins deux fois dans ce code et à sérieusement m'interroger dessus, avec zéro retour.

Tout finit par arriver.

l'url de la demande qui a ajouté le montant dans le panier et a lancé le paiement.

Ouais il manque un point, si on arrive sur `BasketItemPayView` et que `next_url` est vide, alors on prend `item.source_url` pour valeur, i.e. l'URL du formulaire normalement. On a quand même besoin de stocker un `next_url` quelque-part quand on part d'une cellule (on veut revenir sur sa page), mais je pense avoir déjà suggéré de stocker ça dans `Transaction` plutôt qu'en session.

Ok je prends note de ce que vous dites et je vais voir ce qu'on peut faire pour simplifier tout ça comme pour email, l'idée étant que `next_url` ne peut

pas arriver vide dans `handle_payment()`, chaque vue doit savoir où on doit terminer à coup sûr avant d'initier le paiement (c'est plus simple que le jeu de cache cache actuel).