

Combo - Development #42581

Adaptations pour mollie

06 mai 2020 17:40 - Valentin Deniaud

| | | | |
|---|------------------|----------------------|-------------|
| Statut: | Fermé | Début: | 06 mai 2020 |
| Priorité: | Normal | Echéance: | |
| Assigné à: | Valentin Deniaud | % réalisé: | 0% |
| Catégorie: | | Temps estimé: | 0:00 heure |
| Version cible: | | Planning: | Non |
| Patch proposed: | Oui | | |
| Description (plusieurs commits de deux trois lignes et pas très envie de faire autant de tickets) | | | |

Révisions associées

Révision 6009f088 - 06 mai 2020 17:42 - Valentin Deniaud

lingo: add mollie and keyware backends (#42581)

Révision 5d3f080f - 06 mai 2020 17:42 - Valentin Deniaud

lingo: allow overriding automatic_return_url (#42581)

Useful when testing.

Révision a66c7c40 - 11 mai 2020 16:58 - Valentin Deniaud

lingo: catch PaymentException on backend response (#42581)

Révision bf8577f9 - 11 mai 2020 16:58 - Valentin Deniaud

lingo: allow empty payload in ReturnView (#42581)

Révision 31d9b1f9 - 15 juin 2020 12:13 - Valentin Deniaud

lingo: add mollie and keyware backends (#42581)

Révision ea0657f5 - 15 juin 2020 12:13 - Valentin Deniaud

lingo: catch PaymentException on backend response (#42581)

Révision bda15dbc - 15 juin 2020 12:13 - Valentin Deniaud

lingo: handle empty payload in ReturnView (#42581)

Historique

#1 - 06 mai 2020 17:47 - Valentin Deniaud

- Fichier 0001-lingo-add-mollie-and-keyware-backends-42581.patch ajouté
- Fichier 0002-lingo-allow-overriding-automatic_return_url-42581.patch ajouté
- Fichier 0003-lingo-catch-PaymentException-on-backend-response-425.patch ajouté
- Fichier 0004-lingo-allow-empty-payload-in-ReturnView-42581.patch ajouté
- Statut changé de Nouveau à En cours
- Patch proposed changé de Non à Oui

0002 : pour tester le callback, je dois fournir une url résolvable, je fais ça avec un reverse proxy depuis un autre domaine, donc je dois pouvoir mettre l'URL à appeler manuellement dans les settings du backend.

0003 : cette exception n'est pas rattrapée, pourtant il y a déjà des endroits où elle est levée.

0004 : Mollie ne met pas de query string lors de la redirection, c'est normal (entre temps on a été appelé sur le callback et on a le statut du paiement)

Plus de patches à venir, peut-être.

#2 - 07 mai 2020 16:31 - Benjamin Dauvergne

Valentin Deniaud a écrit :

0002 : pour tester le callback, je dois fournir une url résolvable, je fais ça avec un reverse proxy depuis un autre domaine, donc je dois pouvoir mettre l'URL à appeler manuellement dans les settings du backend.
0003 : cette exception n'est pas rattrapée, pourtant il y a déjà des endroits où elle est levée.

Je ne comprends pas bien ce patch vu que :

```
class PaymentException(Exception) :
    pass

class ResponseError(PaymentException) :
    pass
```

0004 : Mollie ne met pas de query string lors de la redirection, c'est normal (entre temps on a été appelé sur le callback et on a le statut du paiement)

On ne peut pas espérer que le callback ait toujours eu lieu avant (problème réseau?), je préférerais que le backend indique qu'il n'attend pas de retour ou alors que backend.response() le gère (si not query_string alors on est sur le redirect et on gère ça d'une manière différente); en tout cas on a la Transaction initiale au retour ainsi que le Payment.id (de mollie) dans Transaction.order_id; peut-être étendre l'interface Payment.response() pour prendre des arguments supplémentaires order_id_hint=xxx, redirect=True qui aidera à trouver le résultat de la réponse dans ces cas là.

#3 - 11 mai 2020 17:06 - Valentin Deniaud

Benjamin Dauvergne a écrit :

Valentin Deniaud a écrit :

0002 : pour tester le callback, je dois fournir une url résolvable, je fais ça avec un reverse proxy depuis un autre domaine, donc je dois pouvoir mettre l'URL à appeler manuellement dans les settings du backend.
0003 : cette exception n'est pas rattrapée, pourtant il y a déjà des endroits où elle est levée.

Je ne comprends pas bien ce patch vu que :

Je n'avais pas remarqué, c'est corrigé dans la branche.

0004 : Mollie ne met pas de query string lors de la redirection, c'est normal (entre temps on a été appelé sur le callback et on a le statut du paiement)

On ne peut pas espérer que le callback ait toujours eu lieu avant (problème réseau?)

Si, Mollie attend qu'on lui ait répondu un 200 avant de rediriger vers le site marchand, j'ai confirmé expérimentalement et ils insistent pas mal dessus dans la doc (<https://docs.mollie.com/payments/overview> et ailleurs).

Après ce code n'est pas spécifique Mollie, tu préfères quand même que Payment.response soit appelé pour les cas d'usage futurs ?

#4 - 11 mai 2020 17:46 - Benjamin Dauvergne

Mais donc en cas de souci réseau direct entre Publik et mollie, le paiement est impossible ? (C'est pas gênant mais ça diffère de la plupart des autres, même PayFiP web-service qui ressemblerait à mollie dans l'esprit n'a pas cette limitation).

PS: en fait si on ne retourne pas 200 ou on ne répond il se passe quoi d'après leur documentation ?

#5 - 11 mai 2020 18:15 - Valentin Deniaud

Désolé je suis imprécis, on est sûrs que le callback a eu lieu mais pas qu'on a réussi à le traiter.

Mollie attend n'importe quelle réponse (pas 200 comme je l'écrivais) ou un timeout de 15s pour rediriger. Ensuite tant qu'on a pas renvoyé 200 on continue à être appelés sur le callback, avec backoff.

Du coup si on veut traiter le cas un peu limite de problème réseau dont tu parles, il faudrait essayer de ne pas court-circuiter toute la logique de callback, donc de ne pas systématiquement appeler sur une redirection. Plutôt passer le statut de la transaction en plus de son ID, laisser Payment.response() décider de ce qu'il fait (ne pas rappeler si le statut est final, ie on a déjà été notifiés).

#6 - 11 mai 2020 18:56 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Du coup si on veut traiter le cas un peu limite de problème réseau dont tu parles, il faudrait essayer de ne pas court-circuiter toute la logique de callback, donc de ne pas systématiquement appeler sur une redirection. Plutôt passer le statut de la transaction en plus de son ID, laisser

Payment.response() décider de ce qu'il fait (ne pas rappeler si le statut est final, ie on a déjà été notifiés).

Je suis un peu perdu, on parle du redirect ici ou du callback web-service ?

La manière dont je vois le fonctionnement actuel : certains backends font des retours usager en redirect avec une requête signée (SIPS il me semble) ou un moyen de valider immédiatement le statut de la transaction via un web-service (PayFiP / mollie), donc dès réception (via PaymentResponse.signed) on peut poser le statut de la transaction, sinon d'autres backends n'envoient le statut que via le callback/webhook web-service ce qui fait qu'on doit attendre sur une page intermédiaire la validation de la transaction.

Qu'est-ce que tu veux dire par "passer le statut de la transaction en plus de son ID, laisser Payment.response() décider de ce qu'il fait" ?

#7 - 12 mai 2020 09:36 - Valentin Deniaud

Si je reprends,

Benjamin Dauvergne a écrit :

On ne peut pas espérer que le callback ait toujours eu lieu avant (problème réseau?), je préférerais que le backend indique qu'il n'attend pas de retour ou alors que backend.response() le gère (si not query_string alors on est sur le redirect et on gère ça d'une manière différente); en tout cas on a la Transaction initiale au retour ainsi que le Payment.id (de mollie) dans Transaction.order_id; peut-être étendre l'interface Payment.response() pour prendre des arguments supplémentaires order_id_hint=xxx, redirect=True qui aidera à trouver le résultat de la réponse dans ces cas là.

Tu me dis que au cas où on ait pas eu le statut de la transaction via le callback, il faut être safe et essayer de le choper quand même dans Payment.response(), au moment de la redirection. Ça implique de rajouter de la logique pour trouver ce order_id_hint, il faut utiliser le transaction_id qu'on a dans l'url de retour, récupérer l'objet Transaction correspondant, et prendre Transaction.order_id.

Ce que je dis c'est que si on passe juste order_id_hint=xxx, redirect=True on se retrouve à toujours appeler Mollie, que le callback ait marché ou pas. Alors que si on passe order_id_hint=xxx, order_status_hint=yyy, redirect=True, on peut regarder si order_status_hint est un statut final, en déduire que le callback a fonctionné et qu'il n'y a pas besoin de réinterroger. C'est une précaution inutile ?

#8 - 12 mai 2020 11:05 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Si je reprends,

Benjamin Dauvergne a écrit :

On ne peut pas espérer que le callback ait toujours eu lieu avant (problème réseau?), je préférerais que le backend indique qu'il n'attend pas de retour ou alors que backend.response() le gère (si not query_string alors on est sur le redirect et on gère ça d'une manière différente); en tout cas on a la Transaction initiale au retour ainsi que le Payment.id (de mollie) dans Transaction.order_id; peut-être étendre l'interface Payment.response() pour prendre des arguments supplémentaires order_id_hint=xxx, redirect=True qui aidera à trouver le résultat de la réponse dans ces cas là.

Tu me dis que au cas où on ait pas eu le statut de la transaction via le callback, il faut être safe et essayer de le choper quand même dans Payment.response(), au moment de la redirection. Ça implique de rajouter de la logique pour trouver ce order_id_hint, il faut utiliser le transaction_id qu'on a dans l'url de retour, récupérer l'objet Transaction correspondant, et prendre Transaction.order_id.

Ce que je dis c'est que si on passe juste order_id_hint=xxx, redirect=True on se retrouve à toujours appeler Mollie, que le callback ait marché ou pas. Alors que si on passe order_id_hint=xxx, order_status_hint=yyy, redirect=True, on peut regarder si order_status_hint est un statut final, en déduire que le callback a fonctionné et qu'il n'y a pas besoin de réinterroger. C'est une précaution inutile ?

Je comprends mieux, merci. Non ça me paraît juste, globalement l'API d'eopayment reste un peu lourdingue (ma faute), on devrait avoir un objet stateful qu'on puisse sérialiser, avec une API claire genre :

```
backend = PaymentBackend(**params)
transaction = backend.new_transaction()
transaction.set_amount(...)
transaction.set_email(...)
transaction.set_description(...)
...
transaction.make_request()
...
serialization = transaction.serialize()
...
transaction = PaymentBackend.deserialize_transaction(serialization)
...
transaction.process_response(...)
...
```

On aurait pas à filer order_id_hint et tout ça, ce serait stocker dans l'objet transaction. En attendant ce que tu proposes est très bien.

#9 - 14 mai 2020 17:31 - Valentin Deniaud

- Fichier 0004-lingo-handle-empty-payload-in-ReturnView-42581.patch ajouté
- Statut changé de En cours à Solution proposée

Voici donc (et branche à jour). Je fais les modifs côté eopayment pour mollie.

#10 - 18 mai 2020 18:05 - Valentin Deniaud

Valentin Deniaud a écrit :

0002 : pour tester le callback, je dois fournir une url résolvable, je fais ça avec un reverse proxy depuis un autre domaine, donc je dois pouvoir mettre l'URL à appeler manuellement dans les settings du backend.

Je vais faire sauter ce patch, avec #6710 ça ne va plus être possible de faire ça. Mais j'aimerais bien le remplacer par quelque chose, ça irait d'ajouter un setting optionnel, qui irait mettre le domaine spécifié à la place du get_absolute_uri ?

#11 - 18 mai 2020 18:20 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Valentin Deniaud a écrit :

0002 : pour tester le callback, je dois fournir une url résolvable, je fais ça avec un reverse proxy depuis un autre domaine, donc je dois pouvoir mettre l'URL à appeler manuellement dans les settings du backend.

Je vais faire sauter ce patch, avec #6710 ça ne va plus être possible de faire ça. Mais j'aimerais bien le remplacer par quelque chose, ça irait d'ajouter un setting optionnel, qui irait mettre le domaine spécifié à la place du get_absolute_uri ?

Je ne sais pas ce que tu appelles un setting optionnel, moi je pense à du monkeypatch sur une variable globale qui ne fera rien en général.

```
# settings.py
import eopayment.mollie
eopayment.mollie.FORCE_CALLBACK_URL = 'what-the-fuck-i-want'
```

Ça peut éventuellement être un truc plus global (dans eopayment pas eopayment.mollie), j'ai juste du mal à voir comment ça marchera dans lingo qui envoie des URLs dynamiques.

PS: c'est peut-être plus simple d'en faire un callback

```
def return_url_cb(url, automatic):
    return 'https://whatever.com/' + urlparse(url).path
eopayment.RETURN_URL_CB = return_url_cb
```

#12 - 18 mai 2020 18:32 - Valentin Deniaud

OK j'ai pas pensé et jamais joué avec monkeypatch hors des tests mais sûrement qu'en patchant dans settings.py ça passe (si c'est ce que tu suggères). Goodbye 0002, branche à jour.

#13 - 08 juin 2020 14:22 - Valentin Deniaud

- Fichier 0002-lingo-catch-PaymentException-on-backend-response-425.patch ajouté
- Fichier 0003-lingo-handle-empty-payload-in-ReturnView-42581.patch ajouté
- Fichier 0001-lingo-add-mollie-and-keyware-backends-42581.patch ajouté

Je remets les patches, j'ai juste simplifié un peu le dernier (qui est en attente de relecture).

#14 - 09 juin 2020 14:21 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

#15 - 15 juin 2020 12:15 - Valentin Deniaud

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

```
commit bda15dbcef75882e1d2e74501f105a43d6cb12ea
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Wed May 6 17:38:05 2020 +0200
```

lingo: handle empty payload in ReturnView (#42581)

commit ea0657f56570bee8dc79d420793d3af11bb15525
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Wed May 6 17:37:22 2020 +0200

lingo: catch PaymentException on backend response (#42581)

commit 31d9b1f96050280f5a70f1920fdaaab9dlb0af6c
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Mar 5 10:15:43 2020 +0100

lingo: add mollie and keyware backends (#42581)

#16 - 15 juin 2020 20:16 - Frédéric Péters

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

Fichiers

| | | | |
|---|------------|--------------|------------------|
| 0001-lingo-add-mollie-and-keyware-backends-42581.patch | 761 octets | 06 mai 2020 | Valentin Deniaud |
| 0002-lingo-allow-overriding-automatic_return_url-42581.patch | 1,21 ko | 06 mai 2020 | Valentin Deniaud |
| 0003-lingo-catch-PaymentException-on-backend-response-425.patch | 1,03 ko | 06 mai 2020 | Valentin Deniaud |
| 0004-lingo-allow-empty-payload-in-ReturnView-42581.patch | 1,33 ko | 06 mai 2020 | Valentin Deniaud |
| 0004-lingo-handle-empty-payload-in-ReturnView-42581.patch | 5,05 ko | 14 mai 2020 | Valentin Deniaud |
| 0002-lingo-catch-PaymentException-on-backend-response-425.patch | 1 ko | 08 juin 2020 | Valentin Deniaud |
| 0003-lingo-handle-empty-payload-in-ReturnView-42581.patch | 5,88 ko | 08 juin 2020 | Valentin Deniaud |
| 0001-lingo-add-mollie-and-keyware-backends-42581.patch | 761 octets | 08 juin 2020 | Valentin Deniaud |