Combo - Development #49149

utiliser la méthode payment_status d'eopayment quand elle est présente pour valider les transactions en tâche de fond

05 décembre 2020 09:18 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	05 décembre 2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Pour PayFiP où on a des soucis pour recevoir les notifications on peut se permettre de récupérer l'état des transactions par polling.

Demandes liées:

Bloqué par EOPayment - Development #49148: faire de payment_status un accesse...	**Fermé**	**05 décembre 2020**

Révisions associées

Révision 99a912e2 - 06 mai 2021 00:31 - Benjamin Dauvergne

lingo: factorize eopayment response handling (#49149)

Révision 5b5d0464 - 06 mai 2021 10:34 - Benjamin Dauvergne

misc: add setting to force synchronous rendering of cells (#49149)

The setting is used through a new fixture, `synchronous_cells`. It's
necessary to test views with cells which always use ajax rendering
without an headless browser or using the request factory.

Révision 760ccb41 - 06 mai 2021 11:04 - Benjamin Dauvergne

lingo: add poll_backend method to PaymentBackend and Transaction (#49149)

Some payment backends in eopayment (like PayFiP) allow polling the
status of currently running transaction, and can signal if a running
transaction has expired. The new can_poll_backend() and poll_backend()
method on Transaction implement this conditional behaviour in lingo.

Historique

#1 - 05 décembre 2020 09:18 - Benjamin Dauvergne

- Bloqué par Development #49148: faire de payment_status un accesseur de la méthode sur le backend ajouté

#2 - 05 décembre 2020 09:18 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#3 - 30 avril 2021 02:33 - Benjamin Dauvergne

- Fichier 0002-lingo-add-poll_backend-method-to-PaymentBackend-and-.patch ajouté

- Fichier 0004-lingo-add-command-to-poll-backends-49149.patch ajouté

- Fichier 0003-lingo-remove-use-of-PaymentBackend.get_payment-49149.patch ajouté

- Fichier 0005-lingo-poll-running-transactions-status-in-cells-4914.patch ajouté

- Fichier 0001-lingo-factorize-eopayment-response-handling-49149.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#4 - 30 avril 2021 12:34 - Thomas Noël

Première relecture...

Dans le hourly, on ne ferait pas self.poll_transaction_status() en premier ? Histoire de notifier aussitôt les paiements réussis à cette occasion.

Sur la commande poll-backends :

- gérer un -v0 pour les print (dans l'idée que ça devienne un cron, cf ci-dessous)
- une typo "--all-baskends" dans poll-backend.py, on gère pas des baskets ici monsieur
- et en mode pénible, je dirais aussi que ça manque de test

Mais plus globalement je me demande si le "hourly" est assez fréquent et je me dis que cette commande "poll-backends" pourrait être renommée et devenir une commande à lancer par un cron général du système de paiement, lancé toutes les 10 minutes et remplaçant complétement le hourly

Voilà en fait toute ma question c'est surtout "est-ce que hourly c'est assez fréquent", j'ai l'impression que non, attendre jusqu'à une heure pour voir finalement son paiement accepté, c'est un peu relou pour un usager, je trouve.

(J'ai relu le reste et pour l'instant pas vu de truc bizarre, hormis les notations genre « payment_backend.backend » mais dans eopayment/lingo on n'est plus à ça près !)

#5 - 30 avril 2021 14:12 - Benjamin Dauvergne

Thomas Noël a écrit :

> Mais plus globalement je me demande si le "hourly" est assez fréquent et je me dis que cette commande "poll-backends" pourrait être renommée et devenir une commande à lancer par un cron général du système de paiement, lancé toutes les 10 minutes et remplaçant complètement le hourly

T'as raté le dernier patch où le poll_backend est fait pour les transactions en cours de l'utilisateur connecté à chaque affichage de la page: mon idée c'est que dans le cas courant, il y a une seule transaction ouverte qui a réussi et dès qu'on revient sur la page des factures (à supposer qu'on ait pas déjà reçu la notification synchrone), ce sera rafraîchi et basta. En fait le poll_backend() en cron n'est là que pour rattraper les cas normalement rares où la personne n'est pas revenu sur le portail normalement. C'est aussi le commit qui contient les tests les plus intéressant je pense (mais si tu as des idées de test que tu aimerais voir dis, de mon coté je vais faire le tour de la couverture du code dans jenkins).

Je réponds plus tard sur le reste.

#6 - 30 avril 2021 14:58 - Thomas Noël

Benjamin Dauvergne a écrit :

> T'as raté le dernier patch où le poll_backend est fait pour les transactions en cours de l'utilisateur connecté à chaque affichage de la page: mon idée c'est que dans le cas courant, il y a une seule transaction ouverte qui a réussi et dès qu'on revient sur la page des factures

Oui j'ai vu ce rafraîchissement dans une cellule et c'est très cool. Mais j'imagine aussi les cas de paiement sans cette cellule (paiement "sans panier"). Mais surtout je voulais juste dire que "hourly" ça me semble pas une fréquence faible pour faire ce genre de check, que je trouve mieux que les choses s'écoulent plus "au fil de l'eau" et donc avoir une fréquence plus élevée me semblait intéressant. Mais ça sera plutôt l'objet d'un autre ticket, évitons de surcharger ici.

Pour mon « ça manque de test » c'est juste que j'avais été voir la couverture, et poll-backends.py est tout rouge roooh.

#7 - 06 mai 2021 00:30 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#8 - 06 mai 2021 15:23 - Benjamin Dauvergne

Thomas Noël a écrit :

> Oui j'ai vu ce rafraîchissement dans une cellule et c'est très cool. Mais j'imagine aussi les cas de paiement sans cette cellule (paiement "sans panier"). Mais surtout je voulais juste dire que "hourly" ça me semble pas une fréquence faible pour faire ce genre de check, que je trouve mieux que les choses s'écoulent plus "au fil de l'eau" et donc avoir une fréquence plus élevée me semblait intéressant. Mais ça sera plutôt l'objet d'un autre ticket, évitons de surcharger ici.

C'est géré aussi, tous les paiements passant par les méthodes de PayMixin (avec ou sans panier), le paiement est bloqué si la transaction est en cours de paiement jusqu'à ce que le statut change de en cours à expiré ou payé (ça prend 20 minutes avec PayFiP de faire le distingo, ou moins si la transaction a été payée). On pourrait en faire le comportement[1] par défaut même quand on ne dispose pas de polling, mais ce sera pour un autre ticket.

> Pour mon « ça manque de test » c'est juste que j'avais été voir la couverture, et poll-backends.py est tout rouge roooh.

J'ai viré toute utilisation de hourly, maintenant ça passe uniquement par la commande qui est lancée toutes les 10 minutes par le cron, par ailleurs je l'ai modifié pour qu'elle puisse s'exécuter de façon concurrente sans problème: je verrouille la transaction lors d'un poll pour un cellule/paiement et pour la commande je ne poll que les transactions verrouillables (deux lancements concurrents vont poller toutes les transactions en se les répartissant automatiquement).

La branche est à jour.

[1] dès qu'on commence un paiement on bloque pendant 20 minutes le paiement sur le/les remote_item/item sélectionnés, sur le site de ma cantine scolaire (Sogeres) ça marche comme ça et je l'ai vu chez Abelium aussi avec PayFiP regie.

#9 - 06 mai 2021 15:23 - Benjamin Dauvergne

- Fichier 0003-lingo-add-poll_backend-method-to-PaymentBackend-and-.patch ajouté

- Fichier 0001-lingo-factorize-eopayment-response-handling-49149.patch ajouté

- Fichier 0002-misc-add-setting-to-force-synchronous-rendering-of-c.patch ajouté

- Statut changé de En cours à Solution proposée

#10 - 06 mai 2021 17:41 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

J'aime bien comme ça ! Allons-y pour envoyer ça en recette illico.

#11 - 06 mai 2021 17:52 - Benjamin Dauvergne

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

```
commit 760ccb41fd16f3907ae0ff921ab6b8d50dd0f5da
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Sat Dec 5 09:24:33 2020 +0100

    lingo: add poll_backend method to PaymentBackend and Transaction (#49149)

    Some payment backends in eopayment (like PayFiP) allow polling the
    status of currently running transaction, and can signal if a running
    transaction has expired. The new can_poll_backend() and poll_backend()
    method on Transaction implement this conditional behaviour in lingo.

commit 5b5d0464146824116ed98ee8946330e128c8b981
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Thu May 6 10:17:30 2021 +0200

    misc: add setting to force synchronous rendering of cells (#49149)

    The setting is used through a new fixture, `synchronous_cells`. It's
    necessary to test views with cells which always use ajax rendering
    without an headless browser or using the request factory.

commit 99a912e2550a7661a5037abcdc4ecc4dbfc25f9f
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Sun Dec 6 08:25:08 2020 +0100

    lingo: factorize eopayment response handling (#49149)
```

#12 - 06 mai 2021 19:17 - Frédéric Péters

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

Fichiers

0002-lingo-add-poll_backend-method-to-PaymentBackend-and-.patch	10,9 ko	30 avril 2021	Benjamin Dauvergne
0004-lingo-add-command-to-poll-backends-49149.patch	3,57 ko	30 avril 2021	Benjamin Dauvergne
0003-lingo-remove-use-of-PaymentBackend.get_payment-49149.patch	3,96 ko	30 avril 2021	Benjamin Dauvergne
0005-lingo-poll-running-transactions-status-in-cells-4914.patch	16,9 ko	30 avril 2021	Benjamin Dauvergne
0001-lingo-factorize-eopayment-response-handling-49149.patch	18,5 ko	30 avril 2021	Benjamin Dauvergne
0003-lingo-add-poll_backend-method-to-PaymentBackend-and-.patch	44,6 ko	06 mai 2021	Benjamin Dauvergne
0001-lingo-factorize-eopayment-response-handling-49149.patch	18,5 ko	06 mai 2021	Benjamin Dauvergne
0002-misc-add-setting-to-force-synchronous-rendering-of-c.patch	1,43 ko	06 mai 2021	Benjamin Dauvergne