Au quotidien - Bug #1264

validation des paiement renforcée

17 février 2012 14:20 - Thomas Noël

Statut:	Fermé	**Début:**	17 février 2012
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	100%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:	Au-quotidien 2012.2		
Patch proposed:		**Planning:**	

Description

~~Dans le patch joint~~, deux choses pour être "sûr" qu'un paiement est bien validé :

- ajout de *regie.validate_on_browser_redirect* : permet à une régie d'accepter la confirmation du paiement sur le redirect final (utile si le callback n'a pas marché) [patch de benjamin]
- on enregistre *invoice.paid=True* même si le formdata associé n'est plus en état d'attente du paiement (en théorie le workflow devrait faire que ça n'arrive jamais, mais je préfère que le paiement soit marqué comme effectué de toutes façons)

Historique

#1 - 17 février 2012 15:06 - Anonyme

Il faut peut-être rajouter dans un commentaire de l'option qu'elle peut potentiellement casser la sécurité du protocole de paiement. Dans le cas de spplus par exemple seule la réponse direct de la banque au commerçant contient une signature, le retour via le navigateur du client n'est pas signé. On ne peut donc pas légitimement le considérer comme une validation du paiement, mais pour débuger localement, c'est quand même plus pratique.

#2 - 17 février 2012 15:16 - Thomas Noël

Anonyme a écrit :

> Il faut peut-être rajouter dans un commentaire de l'option qu'elle peut potentiellement casser la sécurité du protocole de paiement. Dans le cas de spplus par exemple seule la réponse direct de la banque au commerçant contient une signature, le retour via le navigateur du client n'est pas signé. On ne peut donc pas légitimement le considérer comme une validation du paiement, mais pour débuger localement, c'est quand même plus pratique.

Ok. Dans ce cas, est-ce qu'on pourrait déplacer l'option dans eopayment ? Lui pourrait nous dire si un backend de payment assure ou pas la sécurité ?

#3 - 17 février 2012 15:35 - Benjamin Dauvergne

redmine@entrouvert.com écrivait:

> La demande #1264 a été mise à jour par Thomas Noël.
>
> Anonyme a écrit :
>
> > Il faut peut-être rajouter dans un commentaire de l'option qu'elle peut potentiellement casser la sécurité du protocole de paiement. Dans le cas de spplus par exemple seule la réponse direct de la banque au commerçant contient une signature, le retour via le navigateur du client n'est pas signé. On ne peut donc pas légitimement le considérer comme une validation du paiement, mais pour débuger localement, c'est quand même plus pratique.
>
> Ok. Dans ce cas, est-ce qu'on pourrait déplacer l'option dans eopayment ? Lui pourrait nous dire si un backend de payment assure ou pas la sécurité ?

C'est déjà le cas, la convention étant que si payment.response retourne un dernier argument qui vaut None (return_content) alors on a pas une réponse validante. Mais j'avoue qu'on pourrait faire plus explicite.

Mais ça n'a pas de rapport avec la choucroute qui est si je me souviens bien que tu puisses tester avec le module "dummy" en local sur ta machine, le but du module dummy étant aussi de pouvoir tester la validation via le callback, il faut quand même pouvoir jongler entre le

deux.

Bon je le programme ce serait:

- changer le retour de payment.response pour un objet avec les champs suivants:
  - result <- déclaratif
  - signed_result <- sûre
  - return_content <- dans le cas d'un callback ça indique ce qu'il faut renvoyer à la banque
  - transaction_id <- l'id attribué par la banque à la transaction
  - bank_error_code <- message d'erreur de la banque si result == False
  - bank_data <- un dictionnaire avec plein de truc qui changeront selon la banque, à logger puis ignorer

- ajouter au module dummy un paramètre booléen: "only_callback_is_signed" qui permettra de gérer le cas où on veut tester avec ou sans validation via le retour dans le navigateur.

#4 - 17 février 2012 23:20 - Thomas Noël

Ok avec ta proposition de modifier le payment.response.

Si j'ai tout compris ça éviterait d'avoir un "regie.validate_on_browser_redirect", qui présente un risque de mauvaise utilisation (même si on met un warning, c'est tellement un truc de test que je préfère éviter).

Si j'ai toujours compris, ça serait ensuite à nous de mettre en place deux plate-formes dummy, une avec et une sans le retour signé ?

#5 - 17 février 2012 23:20 - Thomas Noël

- Fichier auquo-payment.diff supprimé

#6 - 17 février 2012 23:21 - Thomas Noël

- Description mis à jour

#7 - 17 février 2012 23:25 - Thomas Noël

Benjamin Dauvergne a écrit :

> Mais ça n'a pas de rapport avec la choucroute qui est si je me souviens bien que tu puisses tester avec le module "dummy" en local sur ta machine, le but du module dummy étant aussi de pouvoir tester la validation via le callback, il faut quand même pouvoir jongler entre le deux.

Je précise : je tourne un site de test (dev) sur ma machine, qui n'est pas joignable sur Internet. J'utilise la plateforme dummy déployée par nous, mais le callback ne peut fonctionner. La solution de déployer un "dummy2" avec retour utilisateur signé pourrait me permettre de faire des démos/tests qui valident bien les paiements.

Ce que tu proposes permettrait en plus de régler le cas d'une démo totalement locale (sans accès au net), ça serait très bien aussi.

#8 - 18 février 2012 04:47 - Anonyme

Thomas Noël a écrit :

> Si j'ai toujours compris, ça serait ensuite à nous de mettre en place deux plate-formes dummy, une avec et une sans le retour signé ?

Non en fait ce qui change c'est que pour le backend "dummy" tu as une option qui consiste à considérer tous les messages comme signés ou pas. Ça évite de mélanger cette problématique avec de vrais backends.

Dans mon dernier commit sur au-quo j'ai ajouté le support d'options de backend qui ne soient pas des chaines de caractères pour l'occasion.

#9 - 23 février 2012 12:56 - Thomas Noël

- Statut changé de Nouveau à Résolu (à déployer)

#10 - 11 mars 2013 10:31 - Thomas Noël

- Statut changé de Résolu (à déployer) à Fermé

#11 - 22 juin 2013 21:52 - Frédéric Péters

- % réalisé changé de 0 à 100