

EOPayment - Bug #13562

systempayv2: problème d'encodage des prénoms/noms

12 octobre 2016 10:44 - Serghei Mihai

Statut:	Fermé	Début:	12 octobre 2016
Priorité:	Haut	Echéance:	
Assigné à:	Serghei Mihai	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
<p>Internal Server Error: /lingo/pay Traceback (most recent call last): File "/usr/lib/python2.7/dist-packages/django/core/handlers/base.py", line 111, in get_response response = wrapped_callback(request, *callback_args, **callback_kwargs) File "/usr/lib/python2.7/dist-packages/django/views/generic/base.py", line 69, in view return self.dispatch(request, *args, **kwargs) File "/usr/lib/python2.7/dist-packages/django/views/generic/base.py", line 87, in dispatch return handler(request, *args, **kwargs) File "/usr/lib/python2.7/dist-packages/django/db/transaction.py", line 394, in inner return func(*args, **kwargs) File "/usr/lib/python2.7/dist-packages/combo/apps/lingo/views.py", line 333, in post last_name=request.user.last_name) File "/usr/lib/python2.7/dist-packages/eopayment/__init__.py", line 105, in request return self.backend.request(amount, **kwargs) File "/usr/lib/python2.7/dist-packages/eopayment/systempayv2.py", line 335, in request check_vads(fields) File "/usr/lib/python2.7/dist-packages/eopayment/systempayv2.py", line 189, in check_vads if name in kwargs and not parameter.check_value(kwargs[name]): File "/usr/lib/python2.7/dist-packages/eopayment/systempayv2.py", line 61, in check_value if self.max_length and len(str(value)) > self.max_length: UnicodeEncodeError: 'ascii' codec can't encode character u'\xef' in position 3: ordinal not in range(128)</p>			
Demandes liées:			
Lié à EOPayment - Development #13592: gérer les paramètres en entrée de l'API...		Fermé	13 octobre 2016

Révisions associées

Révision 09f2ec7e - 13 octobre 2016 15:13 - Serghei Mihai

systempayv2: handle properly unicode params (#13562)

Historique

#2 - 12 octobre 2016 11:03 - Serghei Mihai

- Fichier 0001-systempayv2-handle-properly-unicode-params-13562.patch ajouté
- Statut changé de Nouveau à En cours
- Patch proposed changé de Non à Oui

#3 - 12 octobre 2016 11:10 - Serghei Mihai

- Fichier 0001-systempayv2-handle-properly-unicode-params-13562.patch supprimé

#4 - 12 octobre 2016 11:11 - Serghei Mihai

- Fichier 0001-systempayv2-handle-properly-unicode-params-13562.patch ajouté

#5 - 12 octobre 2016 11:11 - Benjamin Dauvergne

Ça gère à la fois de l'UTF-8 et de l'unicode en entrée des APIs comme le #12456 ?

#6 - 12 octobre 2016 11:11 - Benjamin Dauvergne

Pourquoi tu normalises ?

#7 - 12 octobre 2016 11:13 - Serghei Mihai

Pour ne pas me planter sur la longueur du paramètre.

#8 - 12 octobre 2016 11:17 - Serghei Mihai

Benjamin Dauvergne a écrit :

Ça gère à la fois de l'UTF-8 et de l'unicode en entrée des APIs comme le #12456 ?

Tu dois parler d'un autre ticket, j'imagine

#9 - 12 octobre 2016 11:26 - Benjamin Dauvergne

Oui dsl mais je pointais déjà ce ticket dans un mail, [#12546](#).

#10 - 12 octobre 2016 11:28 - Benjamin Dauvergne

Normaliser != encoder... faut choisir, si ça doit sortir en UTF-8 et que la longueur est en nombre d'octets et pas de codepoints il faut mesurer la chaîne encodée, pas normalisé. Tu pourrais expliciter l'objectif ? Pour moi l'objectif c'est de pouvoir passer indifféremment de l'UTF-8 ou de l'unicode en entrée de l'API (à l'exclusion des query string qui sont toujours des octets en UTF-8) et que ça marche.

#11 - 12 octobre 2016 11:28 - Benjamin Dauvergne

Et manipuler en unicode à l'intérieur de la lib.

#12 - 12 octobre 2016 11:55 - Serghei Mihai

- Fichier *0001-systempayv2-handle-properly-unicode-params-13562.patch* ajouté

Ça doit sortir en UTF-8, d'après la doc:

https://payzen.eu/wp-content/uploads/integration/Guide_d_implementation_formulaire_paiement_V2_Payzen_V2.9f.pdf

Patch à jour.

#13 - 12 octobre 2016 17:41 - Benjamin Dauvergne

Ok ça sort en UTF-8 ça n'a aucun rapport avec ce qui est fait là, là c'est la validation que la donnée correspond au format attendu, pour la longueur il faut savoir si la contrainte est en codepoint ou en octets, est-ce écrit dans la doc ?

#14 - 12 octobre 2016 17:47 - Serghei Mihai

Pas de spécification précise là-dessus. Mais de la phrase: "Tous les paramètres ci-dessous contenant des caractères spéciaux devront être encodés et transmis à la plateforme de paiement en UTF-8." je déduis qu'il s'agit d'une contrainte en octets.

#15 - 12 octobre 2016 17:50 - Benjamin Dauvergne

Et donc `len(force_text(value)) > self.max_length` c'est ok pour toi ? Ça mesure bien en octets ?

#16 - 12 octobre 2016 18:06 - Serghei Mihai

- Fichier *0001-systempayv2-handle-properly-unicode-params-13562.patch* ajouté

Misère!

Le patch proposé dans [#12546](#) fait déjà ce boulot. Je viens de l'intégrer.

#17 - 12 octobre 2016 18:16 - Benjamin Dauvergne

Si tu ne lis pas les commentaires que je fais... t'es maintenant faut relire tous les tickets :)

#18 - 13 octobre 2016 15:20 - Serghei Mihai

- Fichier *0001-systempayv2-handle-properly-unicode-params-13562.patch* ajouté

Alors c'est mieux de traiter tous les paramètres en entrée comme unicode et passer en utf-8 uniquement lors de la vérification de la signature. Test mis à jour et testé en vrai sur Payzen. Les prénom/nom avec accents remontent dans Payzen.

#19 - 13 octobre 2016 17:08 - Serghei Mihai

- Lié à *Development #13592: gérer les paramètres en entrée de l'API en unicode* ajouté

#20 - 13 octobre 2016 17:44 - Benjamin Dauvergne

À mon avis c'est pas bon vu que tu ne sais pas si value est de l'unicode ou pas quand tu mesures, mais bon allons y, ack.

#21 - 13 octobre 2016 17:52 - Serghei Mihai

J'ai l'impression que si, request transforme tous ses paramètres en entrée en unicode et appelle check_value

#22 - 13 octobre 2016 18:04 - Benjamin Dauvergne

Quel request ? C'est pas une API Django.

#23 - 13 octobre 2016 18:07 - Serghei Mihai

La méthode request du backend.

#24 - 14 octobre 2016 11:37 - Benjamin Dauvergne

Justement il utilise unicode qui va exploser sur de l'UTF-8 au lieu de force_text, la fonction unicode() il ne faudrait plus s'en servir, c'est l'objet de l'autre ticket que tu as ouvert (mais où les choses ne me semblent pas partir dans ce sens donc je vais faire un commentaire). Ack pour celui-ci.

#25 - 14 octobre 2016 11:37 - Benjamin Dauvergne

Et pour répondre à ta réponse si les value sont de l'unicode len(value) va renvoyer une réponse en codepoint (caractère unicode) et pas en octets comme l'attend systempayv2 (je crois).

#26 - 14 octobre 2016 11:56 - Serghei Mihai

- Statut changé de En cours à Résolu (à déployer)

Ok. J'ai poussé.

```
commit 09f2ec7e3de738c8cc489196c63a64fecabc7777
Author: Serghei Mihai <smihai@entrouvert.com>
Date: Wed Oct 12 10:59:58 2016 +0200
```

```
systempayv2: handle properly unicode params (#13562)
```

#27 - 17 octobre 2016 11:06 - Serghei Mihai

Je viens de tester un paiement vers Payzen avec un prénom en unicode de 63 caractères. Ça passe dans Payzen. Donc la longueur est calculée en codepoints, on a fait une fausse hypothèse.

#28 - 17 octobre 2016 12:13 - Benjamin Dauvergne

Ok mais donc le code est bon là, il mesure en codepoints (il mesure selon le format d'entrée dans la chaîne, maintenant il faut s'assurer que c'est toujours de l'unicode, c'est l'objet de l'autre ticket).

#29 - 26 février 2017 23:43 - Serghei Mihai

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-systempayv2-handle-properly-unicode-params-13562.patch	3,03 ko	12 octobre 2016	Serghei Mihai
0001-systempayv2-handle-properly-unicode-params-13562.patch	2,77 ko	12 octobre 2016	Serghei Mihai
0001-systempayv2-handle-properly-unicode-params-13562.patch	3,6 ko	12 octobre 2016	Serghei Mihai
0001-systempayv2-handle-properly-unicode-params-13562.patch	5,38 ko	13 octobre 2016	Serghei Mihai