

Combo - Development #12637

permettre le paiement anonyme de factures par des tiers

19 juillet 2016 11:25 - Thomas Noël

Statut:	Fermé	Début:	19 juillet 2016
Priorité:	Haut	Echéance:	
Assigné à:	Jean-Baptiste Jaillet	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
On veut permettre le paiement d'une facture depuis une URL non authentifiée.			
Pour cela :			
<ul style="list-style-type: none">• on doit afficher une page qui présente la facture sous une forme minimale : juste le montant et un bouton "payer"• l'URL de cet affichage n'est pas devinable : on y chiffre symétriquement du numéro de la facture• si l'utilisateur est connecté et que la facture lui appartient, on affiche les détails, y compris download du PDF le cas échéant			
Pour le paiement, le payeur doit renseigner son mail avant de cliquer sur "payer" (champ mail pré-rempli avec le mail de l'utilisateur s'il est connecté).			
Demandes liées:			
Lié à Combo - Bug #12669: encrypter les id de factures		Fermé	20 juillet 2016
Lié à Combo - Development #13492: avoir une cellule de déclaration + paiement...		Fermé	06 octobre 2016

Révisions associées

Révision 9779837c - 17 octobre 2016 18:01 - Jean-Baptiste Jaillet

lingo: allow invoices anonymous payment (#12637)

Historique

#1 - 19 juillet 2016 18:51 - Frédéric Péters

Pour y ajouter mes notes,

- on doit afficher une page qui présente la facture sous une forme minimale : juste le montant et un bouton "payer"

Pour l'affichage d'une facture, on a actuellement cette URL/vue :

```
url(r'^lingo/item/(?P<regie_id>[\w,-]+)/(?P<item_id>[\w,-]+)/$',  
    ItemView.as_view(), name='view-item'),
```

La vue est prévue pour juste retourner un bout d'HTML, pour affichage dans une popup. Il faudrait avoir un template minimaliste avec <html> etc. et utiliser celui-ci quand il ne s'agit pas d'une requête via ajax.

- l'URL de cet affichage n'est pas devinable : on y chiffre symétriquement du numéro de la facture

À la place du item_id mentionné dans le bout d'urls.py, on aurait un item_crypto_id (genre), qui pourrait être `base64.b64encode(ARC4.new(SECRET_KEY).encrypt('16-1234')).strip('=')` (expérimentations bienvenues, le truc c'est de fournir une URL qui ne soit pas non plus trop longues).

(et le strip ça fait quelque chose de plus joli, et on peut faire une boucle et tenter d'ajouter un puis deux puis trois = tant qu'on a `binascii.Error: Incorrect padding`)

(il faut aussi faire la même chose pour la vue "download-item-pdf")

- si l'utilisateur est connecté et que la facture lui appartient, on affiche les détails, y compris download du PDF le cas échéant

Pour le paiement, le payeur doit renseigner son mail avant de cliquer sur "payer" (champ mail pré-rempli avec le mail de l'utilisateur s'il est connecté).

Et donc, si le payeur n'est pas le "propriétaire" de la facture il doit y avoir ajout de ce champ mail, et il y a du boulot dans la vue PayView pour gérer tout ça (la récupération de la facture, le paiement avec l'email qui a été renseigné et non pas celui de la personne qui serait logguée, etc.).

#2 - 19 juillet 2016 19:25 - Benjamin Dauvergne

Frédéric Péters a écrit :

À la place du item_id mentionné dans le bout d'urls.py, on aurait un item_crypto_id (genre), qui pourrait être `base64.b64encode(ARC4.new(SECRET_KEY).encrypt('16-1234')).strip('=')` (expérimentations bienvenues, le truc c'est de fournir une URL qui ne soit pas non plus trop longues).

Complètement anecdotique mais tant qu'à faire je préférerais qu'on utilise AES qui est aussi dans pycrypto, on peut même utiliser de l'héxa c'est pas si long (mais bon libre choix du constructeur):

```
import binascii

from Crypto.Cipher import AES
from Crypto.Protocol.KDF import PBKDF2
from Crypto import Random

class DecryptionError(Exception):
    pass

def aes_hex_encrypt(key, data):
    '''Generate an AES key from any key material using PBKDF2, and encrypt data using CFB mode. A
    new IV is generated each time, the IV is also used as salt for PBKDF2.
    '''
    iv = Random.get_random_bytes(2) * 8
    aes_key = PBKDF2(key, iv)
    aes = AES.new(aes_key, AES.MODE_CFB, iv)
    crypted = aes.encrypt(data)
    return '%s%s' % (binascii.hexlify(iv[:2]), binascii.hexlify(crypted))

def aes_hex_decrypt(key, payload, raise_on_error=True):
    '''Decrypt data encrypted with aes_base64_encrypt'''
    try:
        iv, crypted = payload[:4], payload[4:]
    except (ValueError, TypeError):
        if raise_on_error:
            raise DecryptionError('bad payload')
        return None
    try:
        iv = binascii.unhexlify(iv) * 8
        crypted = binascii.unhexlify(crypted)
    except TypeError:
        if raise_on_error:
            raise DecryptionError('incorrect hexadecimal encoding')
        return None
    aes_key = PBKDF2(key, iv)
    aes = AES.new(aes_key, AES.MODE_CFB, iv)
    return aes.decrypt(crypted)

invoice_id = '12-1234'
assert aes_hex_decrypt('xxx', aes_hex_encrypt('xxx', invoice_id)) == invoice_id

print aes_hex_encrypt('xxx', invoice_id)

edd4e9cddea211b037
```

#3 - 19 juillet 2016 20:20 - Frédéric Péters

Oui, peu importe la crypto ici.

#4 - 20 juillet 2016 11:31 - Jean-Baptiste Jaillet

Ok j'ai regardé. On se posait la question avec Thomas hier.

Pour le template, je peux récupérer les thèmes quelque part pour que le truc reste cohérent graphiquement?

Et pour le chiffrage, l'exemple que montre Benj, c'est tout ce qu'il faut faire pour chiffrer l'adresse?

#5 - 20 juillet 2016 11:35 - Benjamin Dauvergne

Jean-Baptiste Jaillet a écrit :

Ok j'ai regardé. On se posait la question avec Thomas hier.

Pour le template, je peux récupérer les thèmes quelque part pour que la truc reste cohérent graphiquement?

Et pour le chiffrement, l'exemple que montre Benj, c'est tout ce qu'il faut faire pour chiffrer l'adresse?

Chiffrement.

Tu peux partir sur le code de Fred aussi, faut juste écrire la procédure de déchiffrement, vraiment tu fais comme tu le sens, c'est l'occasion de jouer avec pycrypto c'est tout, on est de toute façon plus sûr de l'obfuscation que de la sécurité ici, on pourrait faire du rot13 au pire.

#6 - 20 juillet 2016 11:41 - Frédéric Péters

Pour le template, je peux récupérer les thèmes quelque part pour que la truc reste cohérent graphiquement?

On n'a pas de template "page vierge", on a un peu de style pour la popup de facture, tu peux juste taper le contenu dans un `<div class="invoice">` et ça pourra être stylé en étendant à `div.invoice` le style qu'on définit actuellement uniquement pour `div.ui-dialog #item`.

Mais je ne suis pas sûr d'avoir bien capté la question.

#7 - 20 juillet 2016 12:14 - Jean-Baptiste Jaillet

Ça répond à la question.

#8 - 20 juillet 2016 14:10 - Thomas Noël

Frédéric Péters a écrit :

Pour l'affichage d'une facture, on a actuellement cette URL/vue :

[...]

La vue est prévue pour juste retourner un bout d'HTML, pour affichage dans une popup. Il faudrait avoir un template minimaliste avec `<html>` etc. et utiliser celui-ci quand il ne s'agit pas d'une requête via ajax.

Discuté hier avec jibé, selon moi il faut faire un affichage qui colle avec celui de la ville. Parce qu'il s'agit d'une page qui va être affichée pour payer une facture, il faut donc que l'utilisateur "voit" bien où il est.

[Un moment, j'imaginai que c'est au niveau de la régie qu'on indiquerait le template de page à utiliser (parmi `settings.COMBO_PUBLIC_TEMPLATES`) mais c'est un peu une idée en l'air]

#9 - 20 juillet 2016 15:00 - Frédéric Péters

Ça ne peut pas être la vue générale; mais que le template "vide" généré contienne quelques `<div>`, genre header, et `publik-base-theme` mettra un bandeau et cie. Mais c'est du taf pour `publik-base-theme`, ici dans `combo` même, avoir un template des plus vides, ça ne pose pas de problème.

#10 - 20 juillet 2016 18:24 - Jean-Baptiste Jaillet

<https://dev.entrouvert.org/issues/12669>

Première petite partie faite, j'ai créé le ticket du coup. C'est passé de l'id clair à encrypté.

#11 - 20 juillet 2016 18:35 - Frédéric Péters

- Lié à Bug #12669: *encrypter les id de factures ajoutés*

#12 - 20 juillet 2016 18:35 - Frédéric Péters

- *Projet changé de Combo à Lingo*

- *Sujet changé de lingo: affichage d'une facture via une URL avec "code secret" à affichage d'une facture via une URL avec "code secret"*

#13 - 20 juillet 2016 18:36 - Frédéric Péters

- *Sujet changé de affichage d'une facture via une URL avec "code secret" à permettre le paiement anonyme de factures par des tiers*

#14 - 22 juillet 2016 18:12 - Jean-Baptiste Jaillet

- *Fichier 0001-lingo-create-view-for-single-item-when-user-s-not-au.patch ajouté*

Ajout de la vue avec l'url et possibilité pour une personne non connectée d'accéder à celle-ci.

Ajout du champ mail pour personne non connectée. Avec Thomas on avait pensé pré-rempli mais comme on récupère directement la valeur de l'utilisateur s'il est connecté, je trouve ça plus cohérent de ne pas avoir ce champ quand un utilisateur est connecté.

#15 - 22 juillet 2016 18:52 - Frédéric Péters

```
@@ -163,7 +163,10 @@ class Regie(models.Model):
    item = requests.get(self.signed_url(request, url,
        NameID=mellon['name_id_content'])).json()
    return build_remote_item(item.get('data'), self)
-    return {}
+    else:
+        url = self.webservice_url + '/invoice/%s/' % item
+        item = requests.get(self.signed_url(request, url)).json()
+        return build_remote_item(item.get('data'), self)
```

Cela exige désormais des connecteurs qu'ils fonctionnent sans paramètre NameID; du coup, on peut juste avoir ce nouveau code, virer la condition et l'autre situation.

```
<div id="content">
- <div id="appbar">
[...]
```

Non, il faut garder le `<div id="appbar">` directement sous le `<div id="content">`.

```
--- a/combo/apps/lingo/templates/lingo/combo/item.html
+++ b/combo/apps/lingo/templates/lingo/combo/item.html
```

Je comprends la motivation à indenter mais en l'espèce ça rend la lecture du patch bien compliquée.

Il me semble que la partie ajoutée, ce sont ces lignes :

```
+         {% if request.user.is_anonymous %}
+         <div class="mail">
+             <label for="mail">{% trans 'Email address' %}</label>
+             <input type="email" id="email" name="email" value="" />
+         </div>
+         {% endif %}
```

address, deux d en anglais. j'y mettrais aussi un attribut required, pour la validation HTML5. Et pas besoin de mettre value vide.

```
else:
    transaction.user = None
+    email = request.POST.get('email')
```

Même après le required ajouté, il faut vérifier qu'on a bien une valeur pour email. (le plus facile étant sans doute d'utiliser ici `messages.warning`)

```
--- a/combo/apps/lingo/templates/lingo/combo/item.html
+++ b/combo/apps/lingo/templates/lingo/combo/item.html
@@ -1,46 +1,55 @@
 {% load i18n %}
 <div id="content">
```

L'affichage d'une facture doit désormais donner une page réelle, avec doctype et `<html>` etc. Revenant sur ce point, regardant ce qu'on fait à Orléans (ex: <https://portail.famille.orleans.fr/facture/simple/tipi/1502044313/cc13259>), je vois qu'on utilisait simplement le style du site, on ne faisait en fait pas de style particulier. J'écrivais : « Il faudrait avoir un template minimaliste avec `<html>` etc. et utiliser celui-ci quand il ne s'agit pas d'une requête via ajax » et personne ne m'a repris là-dessus, mais en fait, donc, cette idée de template minimaliste, elle pourrait être zappée. Lors d'une requête ajax on continue à fournir comme maintenant, et lors d'une requête "pure", on affiche la facture dans une page du site.

Il faudrait aussi, je pense, distinguer la situation de paiement par un tiers pour après le paiement revenir sur la page d'affichage de la facture.

#16 - 25 juillet 2016 18:15 - Jean-Baptiste Jaillet

Il y a quelques points que je suis pas sûr à 100 % du résultat :

- pour le changement des name id, je fais uniquement ce qu'il y dans mon else du coup ? et ensuite on modifie passerelle ou il faut le faire avec ce patch?

```
--- a/combo/apps/lingo/templates/lingo/combo/item.html
+++ b/combo/apps/lingo/templates/lingo/combo/item.html
```

Je comprends la motivation à indenter mais en l'espèce ça rend la lecture du patch bien compliquée.

=> tu parles de quelle indentation? dans le code html?

-pour l'histoire du doctype, est ce que du coup je fais un autre template avec le doctype et en fonction de l'appel je renvoie vers tel ou tel template, ou est ce que la conclusion du sur cette partie c'est on met un doctype parce que ça change rien pour l'appel ajax ?

Pour le warning de l'email par contre pour moi s'il y a pas d'email on doit renvoyer sur la page et pas lancer la facture, mais je ne suis pas sûr de ce point non plus (mais il me semble que l'idée de mettre un warning c'est pas pour annuler l'action).

#17 - 25 juillet 2016 18:50 - Frédéric Péters

Jean-Baptiste Jaillet a écrit :

Il y a quelques points que je suis pas sûr à 100 % du résultat :

- pour le changement des name id, je fais uniquement ce qu'il y a dans mon else du coup ? et ensuite on modifie passerelle ou il faut le faire avec ce patch?

Il faut vérifier les différents connecteurs qui retournent des factures et voir ce qu'ils font du NameID.

- [...] => tu parles de quelle indentation? dans le code html?

Oui, qui fait qu'il a fallu creuser pour sortir les six lignes ajoutées, sans pour autant être sûr qu'il n'y ait eu que ça.

-pour l'histoire du doctype, est ce que du coup je fais un autre template avec le doctype et en fonction de l'appel je renvoie vers tel ou tel template, ou est ce que la conclusion du sur cette partie c'est on met un doctype parce que ça change rien pour l'appel ajax ?

Pour moi, oui, gardons le snippet html pour l'appel ajax, et quand ce n'est pas un appel ajax utilisons un autre template.

Pour le warning de l'email par contre pour moi s'il y a pas d'email on doit renvoyer sur la page et pas lancer la facture, mais je ne suis pas sûr de ce point non plus (mais il me semble que l'idée de mettre un warning c'est pas pour annuler l'action).

Oui, s'il n'y a pas d'email, on fait le messages.warning et on renvoie sur la page d'où on vient.

#18 - 26 juillet 2016 16:33 - Jean-Baptiste Jaillet

- Fichier 0001-lingo-create-view-for-single-item-when-user-s-not-au.patch ajouté

Ok donc j'ai fait les modifs. Juste deux trucs :

-pour le nameID, de ce que j'ai vu dans passerelle pour le stubinvoice, on ne s'en sert pas, c'est le passage dans Requests pour la session mellon qui s'en sert afin de construire l'url.

-pour le template du coup, j'ai ajouté message, mais c'est pas très joli et je sais que c'est géré par gadjo mais je pense du coup que ça serait fait en amont. Donc éventuellement virer mon bloc message que j'ai ajouté.

#19 - 26 juillet 2016 17:10 - Frédéric Péters

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

i> -pour le nameID, de ce que j'ai vu dans passerelle pour le stubinvoice, [...]

Il faut aussi/surtout regarder les vrais connecteurs (agoraplus et teamnet_axel).

-pour le template du coup, j'

Garde item.html comme étant la version "ajax"; ça permet de ne pas s'assurer que nulle part on n'a un template qui override celui-ci et se mettrait à ne plus fonctionner.

Nomme plutôt le nouveau invoice_fullpage.html (qu'on commence à dégager les "item" qui ne veulent rien dire de précis).

Tu pourrais mettre un <title> avec "facture numéro xx".

Mets aussi une classe sur le <body>, pour permettre aux CSS de viser juste.

Il y aurait moyen de faire un include du template "ajax", histoire de ne pas dupliquer son code ?

```
return HttpResponseRedirect(request.META.get('HTTP_REFERER'))
```

Cet entête n'est pas nécessairement présent, il faut assurer la situation où il n'existe pas.

#20 - 28 juillet 2016 14:25 - Jean-Baptiste Jaillet

- Fichier `0001-lingo-create-view-for-single-item-when-user-s-not-au.patch` ajouté

Voilà, j'ai ajouté une classe sur le body, je sais pas si y'en avait une spécifique. Pour le include, le problème c'est que l'ajout des champs mail est en plein milieu.

#21 - 28 juillet 2016 14:29 - Jean-Baptiste Jaillet

- Fichier `0001-lingo-create-view-for-single-item-when-user-s-not-au.patch` ajouté

Je m'étais embrouillé sur les tests de `request.META.get('HTTP_REFERER')`.

#22 - 28 juillet 2016 15:11 - Frédéric Péters

Pour le include, le problème c'est que l'ajout des champs mail est en plein milieu.

On pourrait imaginer qu'une variable soit posée dans le contexte pour que ça s'affiche uniquement dans une situation ?

#23 - 28 juillet 2016 18:30 - Jean-Baptiste Jaillet

- Fichier `0001-lingo-create-view-for-single-item-when-user-s-not-au.patch` ajouté

Fait. Je faisais le test uniquement si la personne était connectée mais j'ai rajouté un `is_ajax`. C'est vraiment de la sécu parce que je pense que l'appel ajax n'est fait que par une personne connectée.

Mais bon au moins on est sûr.

#24 - 29 août 2016 14:42 - Frédéric Péters

Tu peux attacher un patch rebasé sur master ?

#25 - 27 septembre 2016 16:36 - Jean-Baptiste Jaillet

- Fichier `0001-lingo-create-view-for-single-item-when-user-s-not-au.patch` ajouté

Voilà

#26 - 27 septembre 2016 18:42 - Frédéric Péters

`{% blocktrans %}` Facture numéro `{{ number }}` `{% endblocktrans %}` faut de l'anglais.

```
<script>
(function () {
  var $ = jQuery;
  $(document).ready(function() {
    $('.messages').delay(5000).fadeOut('slow');
  });
})();
</script>
```

Oublie l'idée d'un truc de ce genre ici, c'est une vue minimale et le template sera de toute façon revu dans `publik-base-theme`. (et jquery n'est pas présent ici).

`<label for="mail">{% trans 'Email address' %}</label>`, plutôt 'Email:' (le : et en plus ça permet de gagner une traduction déjà présente).

```
if not request.POST.get('email'):
    messages.warning(request, _(u'You must give an email address.'))
    if request.META.get('HTTP_REFERER'):
        return HttpResponseRedirect(request.META.get('HTTP_REFERER'))
```

Plutôt que jouer sur `HTTP_REFERER` (où je dirai toujours qu'il ne faut pas), inclure l'URL de retour dans le formulaire ? (il y a déjà un `next_url` prévu pour).

Alternativement, en son absence, il y a moyen de reconstituer l'URL de la facture vu qu'on a la régie et le numéro de facture.

```
except DecryptionError as e:
    return Http404(str(e))
```

Il me semble que [#12669](#) gère déjà ça.

Tests.

#27 - 06 octobre 2016 12:52 - Frédéric Péters

- Lié à Development #13492: avoir une cellule de déclaration + paiement de facture ajouté

#28 - 06 octobre 2016 12:52 - Frédéric Péters

- Priorité changé de Normal à Haut

#29 - 12 octobre 2016 16:09 - Jean-Baptiste Jaillet

- Fichier 0001-lingo-create-view-for-single-item-when-user-s-not-au.patch ajouté

Voilà.
J'ai un doute sur les tests, je pense que je couvre les nouveautés (mais y'a peut être des trucs dégueulasse sur mes .join()).
Pour l'exception de decryptage comme le ticket est barré j'ai supposé que c'était bon (ou alors c'est son statut dans quel cas je dégage le try).

#30 - 12 octobre 2016 16:52 - Frédéric Péters

Oui, pour les join, nettement, `'/'.join(['/lingo/item', str(regie.pk), str(encrypt_id)]) + '/'`. → `"/lingo/item/%s/%s/" % (regie.pk, encrypt_id)`.

(successful, pas deux l)

`b_item = BasketItem.objects.create(user=user, regie=regie, subject='item', amount='10.5', source_url='/item/1')` je mettrais un XXX ou autre truc nettement différenciant comme `source_url`, pour être sûr de ne pas confondre plus tard le `/item/1` qui viendrait de là.

`resp = client.get(reverse('view-item', kwargs={'item_crypto_id': encrypt_id, 'regie_id': regie.pk}))`

Après ça, tu dois utiliser l'objet `resp` pour vérifier son contenu, présence d'un formulaire, présence de tel et tel champs, et puis utiliser `resp.form.submit(...)` pour poster le formulaire; pas faire un `client.post` indépendant. (qui m'étonne, avec le `csrf_token` présent dans le formulaire)

Le code que tu protèges d'un except `DecryptionError` as e., je me demande comment il pourrait lever cette exception.

#31 - 13 octobre 2016 15:08 - Jean-Baptiste Jaillet

- Fichier 0001-lingo-create-view-for-single-item-when-user-s-not-au.patch ajouté

Voilà.

#32 - 13 octobre 2016 15:58 - Frédéric Péters

```
-         except DecryptionError:
-             raise Http404()
+         except DecryptionError as e:
+             return Http404()
```

Tu gagnes un test sur la présence d'un mauvais id chiffré à écrire.

```
if self.request.is_ajax():
    self.template_name = 'lingo/combo/item.html'
```

Plutôt ajouter une méthode `get_template_names`.

```
{% if not user.is_authenticated and not is_ajax %}
```

Pourquoi le "and not is_ajax" ?

#33 - 14 octobre 2016 02:37 - Jean-Baptiste Jaillet

- Fichier 0001-lingo-create-view-for-single-item-when-user-s-not-au.patch ajouté

En effet, le ajax n'a aucun sens.
Pour le `get_template_names` j'ai un doute, je suis aller voir un peu la doc et ce qu'on fait des gens, et comme on appelle la même fonction et le même objet, je n'ai pas vu d'autres solutions. Il y en a peut être une plus élégante.

Pour le test je pense que c'est bon : je suis passé par parce qu'après une longue prise de tête il semble qu'une 404 avec webtest ba ça déclenche une erreur => tu peux pas tester `status_code = 404`, il n'attends que du 200 / 3xx. Donc je suis passé par client pour ce test.

#34 - 14 octobre 2016 09:38 - Frédéric Péters

Avec WebTest, tu peux faire `.get(..., status=404)` pour tester

#35 - 14 octobre 2016 09:58 - Jean-Baptiste Jaillet

- Fichier 0001-lingo-create-view-for-single-item-when-user-s-not-au.patch ajouté

Voilà, fait la modif, c'est plus "cohérent".

#36 - 17 octobre 2016 17:09 - Frédéric Péters

Le message de commit est un peu pourri; quand le ticket est bien fait, il décrit la fonctionnalité attendue et il suffit de la traduire. "lingo: add support for paying invoices anonymously", genre.

Pour le `get_template_names` j'ai un doute, je suis aller voir un peu la doc et ce qu'on fait des gens, et comme on appelle la même fonction et le même objet, je n'ai pas vu d'autres solutions. Il y en a peut être une plus élégante.

C'est très bien ainsi.

```
-         raise Http404()
+         raise Http404(_('Invoice encryptage error'))
```

Tu peux soit utiliser de l'anglais, soit remettre comme avant.

#37 - 17 octobre 2016 17:30 - Jean-Baptiste Jaillet

- Fichier 0001-cmis-add-cmis-connector-to-upload-file-12876.patch ajouté

Voilà.

#38 - 17 octobre 2016 17:31 - Frédéric Péters

C'est le connecteur CMIS.

#39 - 17 octobre 2016 17:33 - Jean-Baptiste Jaillet

- Fichier 0001-cmis-add-cmis-connector-to-upload-file-12876.patch supprimé

#40 - 17 octobre 2016 17:33 - Jean-Baptiste Jaillet

- Fichier 0001-lingo-allows-invoices-anonymous-payment-12637.patch ajouté

Damn.

#41 - 17 octobre 2016 17:38 - Frédéric Péters

Dans le message de commit, allow, pas allows.

Et un truc que j'avais zappé, faut taper le DOCTYPE tout en haut, sans même de retour à la ligne, `{% load i18n %}<!DOCTYPE html>`.

```
{% blocktrans %} Invoice number {{ number }} {% endblocktrans %}
```

Virer les espaces autour.

Avec ces trois changements, ack.

#42 - 17 octobre 2016 18:10 - Jean-Baptiste Jaillet

- Fichier 0001-cmis-add-cmis-connector-to-upload-file-12876.patch ajouté

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

Hop. C'est poussé.

#43 - 17 octobre 2016 18:13 - Jean-Baptiste Jaillet

- Fichier 0001-cmis-add-cmis-connector-to-upload-file-12876.patch supprimé

#44 - 17 octobre 2016 18:13 - Jean-Baptiste Jaillet

- Fichier 0001-lingo-allow-invoices-anonymous-payment-12637.patch ajouté

...

#45 - 23 décembre 2018 15:24 - Frédéric Péters

Fichiers

0001-lingo-create-view-for-single-item-when-user-s-not-au.patch	6,73 ko	22 juillet 2016	Jean-Baptiste Jaillet
0001-lingo-create-view-for-single-item-when-user-s-not-au.patch	9,88 ko	26 juillet 2016	Jean-Baptiste Jaillet
0001-lingo-create-view-for-single-item-when-user-s-not-au.patch	7,81 ko	28 juillet 2016	Jean-Baptiste Jaillet
0001-lingo-create-view-for-single-item-when-user-s-not-au.patch	7,96 ko	28 juillet 2016	Jean-Baptiste Jaillet
0001-lingo-create-view-for-single-item-when-user-s-not-au.patch	6,27 ko	28 juillet 2016	Jean-Baptiste Jaillet
0001-lingo-create-view-for-single-item-when-user-s-not-au.patch	6,49 ko	27 septembre 2016	Jean-Baptiste Jaillet
0001-lingo-create-view-for-single-item-when-user-s-not-au.patch	9,75 ko	12 octobre 2016	Jean-Baptiste Jaillet
0001-lingo-create-view-for-single-item-when-user-s-not-au.patch	10,1 ko	13 octobre 2016	Jean-Baptiste Jaillet
0001-lingo-create-view-for-single-item-when-user-s-not-au.patch	10,6 ko	14 octobre 2016	Jean-Baptiste Jaillet
0001-lingo-create-view-for-single-item-when-user-s-not-au.patch	10,5 ko	14 octobre 2016	Jean-Baptiste Jaillet
0001-lingo-allows-invoices-anonymous-payment-12637.patch	10,4 ko	17 octobre 2016	Jean-Baptiste Jaillet
0001-lingo-allow-invoices-anonymous-payment-12637.patch	10,4 ko	17 octobre 2016	Jean-Baptiste Jaillet