w.c.s. - Development #35013

Implémenter un filtre pour l'algorithme de Luhn

23 juillet 2019 17:20 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	23 juillet 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Cela permettra de valider les numéros de SIREN, SIRET, CB. (Pas le numéro de sécu qui utilise une autre formule, http://marlot.org/util/calcul-de-la-cle-nir.php).

Demandes liées:

Lié à w.c.s. - Development #30849: Filtres Django pour la validation des champs	**Rejeté**	**22 février 2019**
Lié à w.c.s. - Development #11455: Évolution de l'option "regex de validation"	**Fermé**	**21 juin 2016**

Révisions associées

Révision b7cbff13 - 13 août 2019 15:10 - Benjamin Dauvergne

fields: add Luhn algorithm to string field validation (#35013)

Historique

#1 - 23 juillet 2019 17:22 - Frédéric Péters

- Lié à Development #30849: Filtres Django pour la validation des champs ajouté

#2 - 23 juillet 2019 17:40 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#3 - 23 juillet 2019 17:42 - Benjamin Dauvergne

- Fichier 0001-template-filter-for-Luhn-algorithm-35013.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Testé sur les numéros SIRET/SIREN d'EO et leurs variations fausses.

#4 - 23 juillet 2019 18:13 - Benjamin Dauvergne

- Lié à Development #11455: Évolution de l'option "regex de validation" ajouté

#5 - 23 juillet 2019 21:44 - Frédéric Péters

Les changements six.text_type ne sont pas souhaités ici (et |luhn lui-même ne devrait pas utiliser ça; quand l'heure sera venue de s'occuper des caractères/unicode dans le passage à python 3, je préférerai ne pas devoir m'interroger sur le pourquoi le code serait différent ici).

Il faudra un nom plus explicite que |luhn, |is_valid_xxx, en multipliant les _xxx, ou plus conservateur |has_valid_xxx_control ?

Dans #30849 un frein avait été mis sur ces filtres, en pointant plutôt #11455 (de mon côté je peux laisser glisser ça).

#6 - 24 juillet 2019 06:27 - Benjamin Dauvergne

- Fichier 0001-template-filter-for-Luhn-algorithm-35013.patch ajouté

Remarques intégrés, is_valid_luhn (non exposé) exposé sous les deux noms is_valid_siret et is_valid_siren.

#7 - 24 juillet 2019 06:27 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Dans #30849 un frein avait été mis sur ces filtres, en pointant plutôt #11455 (de mon côté je peux laisser glisser ça).

Je pense qu'on peut vivre avec les deux, il sera toujours temps au niveau documentation de favoriser l'un ou l'autre.

#8 - 24 juillet 2019 09:39 - Thomas Noël

Pour is_valid_siren et siret, tu ne peux pas ajouter une vérification de la longueur ? Et, pendant qu'on y est, laisser passer tous les SIRET de la Poste (356000000xxxxx, comme tu as remarqué https://fr.wikipedia.org/wiki/Formule_de_Luhn#Cas_particulier)

#9 - 24 juillet 2019 13:49 - Benjamin Dauvergne

- Fichier 0001-template-filter-for-Luhn-algorithm-35013.patch ajouté

Voilà.

#10 - 24 juillet 2019 13:58 - Thomas Noël

Je me demande si on devrait pas exposer clean_number et si les gens veulent valider des trucs bizarres (non validés par une regex genre \d+) alors ils doivent faire un form_var_truc|clean_number|is_valid_siren

Dans l'idée que si ensuite tu dois appeler un webservice avec le siren en argument, ça se fera avec form_var_truc|clean_number (parce que si on n'a pas le clean_number ça sera pas marrant)

#11 - 24 juillet 2019 15:53 - Benjamin Dauvergne

Thomas Noël a écrit :

> Je me demande si on devrait pas exposer clean_number et si les gens veulent valider des trucs bizarres (non validés par une regex genre \d+) alors ils doivent faire un form_var_truc|clean_number|is_valid_siren
>
> Dans l'idée que si ensuite tu dois appeler un webservice avec le siren en argument, ça se fera avec form_var_truc|clean_number (parce que si on n'a pas le clean_number ça sera pas marrant)

Ok, mais c'est pour ça que je n'aime pas les conditions tels qu'elles sont faites, dans un outil de formulaire on doit valider et normaliser en même temps (clairement on devrait systématiquement faire des strip() par exemple, virer les caractères unicode non graphiques).

#12 - 24 juillet 2019 16:07 - Thomas Noël

Benjamin Dauvergne a écrit :

> Ok, mais c'est pour ça que je n'aime pas les conditions tels qu'elles sont faites, dans un outil de formulaire on doit valider et normaliser en même temps (clairement on devrait systématiquement faire des strip() par exemple, virer les caractères unicode non graphiques).

Je sais et je suis d'accord mais ça c'est plutôt #11455 (étendre la notion de validation, et pourquoi pas y inclure un clean ensuite)... on n'en est pas là.

#13 - 24 juillet 2019 16:59 - Benjamin Dauvergne

- Fichier 0002-template-filter-for-Luhn-algorithm-35013.patch ajouté

- Fichier 0001-template-filter-cleaning-numbers-35013.patch ajouté

Voilà.

#14 - 24 juillet 2019 17:13 - Benjamin Dauvergne

- Fichier 0001-template-filter-for-Luhn-algorithm-35013.patch ajouté

Dernière version je l'espère, en renvoyant soit '' soit la chaîne valide et normalisée, les validateurs servent aussi à normaliser, ça évite d'exposer un clean_number un peu moche.

#15 - 24 juillet 2019 17:52 - Frédéric Péters

Vraiment je préfère que les is_whatever renvoient vrai/faux.

s/couting/counting/ dans un commentaire. Pas d'espace avant : en anglais (commentaires "special case : La Poste"). Pour la partie La Poste, être complet et coder la vérification supplémentaire ("somme simple des chiffres du SIRET est un multiple de cinq") pour le SIRET ?

#16 - 24 juillet 2019 19:52 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Vraiment je préfère que les is_whatever renvoient vrai/faux.

De mon coté je n'ai pas envie d'exposer clean_number c'est pas documentable/utile. Donc un clean_siren, clean_siret en plus ?

> s/couting/counting/ dans un commentaire. Pas d'espace avant : en anglais (commentaires "special case : La Poste"). Pour la partie La Poste, être complet et coder la vérification supplémentaire ("somme simple des chiffres du SIRET est un multiple de cinq") pour le SIRET ?

Ok.

#17 - 26 juillet 2019 14:22 - Frédéric Péters

- Sujet changé de Implémenter un filtre pour l'agorithme de Luhn à Implémenter un filtre pour l'algorithme de Luhn

> De mon coté je n'ai pas envie d'exposer clean_number c'est pas documentable/utile. Donc un clean_siren, clean_siret en plus ?

Je dirais que non, qu'il faut introduire le moins de choses ici, juste la validation, que le format doit pour le moment être forcé par une regex sur le champ, que plus avancé ça devra être #11455.

#19 - 10 août 2019 18:35 - Benjamin Dauvergne

- Fichier 0001-fields-add-Luhn-algorithm-to-string-field-validation.patch ajouté

Patch reconstruit au dessus de #11455.

#20 - 13 août 2019 14:20 - Thomas Noël

Il faut vérifier que les int(x) vont pas planter si x n'est pas un chiffre, genre là avec ce test :

```
--- a/tests/test_widgets.py
+++ b/tests/test_widgets.py
@@ -668,6 +668,12 @@ def test_wcsextrastringwidget_siret_validation():
     mock_form_submission(req, widget, {'test': '44317013900037'})
     assert widget.has_error()

+    # failing case
+    widget = WcsExtraStringWidget('test', value='foo', required=False)
+    widget.field = fakefield
+    mock_form_submission(req, widget, {'test': 'ABC17013900037'})
+    assert widget.has_error()
+
```

ça fait boum :

```
>               checksum += sum(int(y) for y in str(2 * int(x)))
E               ValueError: invalid literal for int() with base 10: 'C'
```

J'ai envoyé un commit dans la branche qui fait du isdigit()... à merger avec ton patch si tu es ok, puis pousser le patch résultat.

#21 - 13 août 2019 15:10 - Benjamin Dauvergne

- Fichier 0001-fields-add-Luhn-algorithm-to-string-field-validation.patch ajouté

Voilà mergé pushé, yapuka ack.

#22 - 13 août 2019 15:11 - Benjamin Dauvergne

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

```
commit b7cbff134c933b5cbe9808d3e9a305ee255c35c3
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Tue Jul 23 17:40:14 2019 +0200

    fields: add Luhn algorithm to string field validation (#35013)
```

#23 - 14 août 2019 11:15 - Frédéric Péters

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

Fichiers

0001-template-filter-for-Luhn-algorithm-35013.patch	3,9 ko	23 juillet 2019	Benjamin Dauvergne
0001-template-filter-for-Luhn-algorithm-35013.patch	3,15 ko	24 juillet 2019	Benjamin Dauvergne
0001-template-filter-for-Luhn-algorithm-35013.patch	4,66 ko	24 juillet 2019	Benjamin Dauvergne
0002-template-filter-for-Luhn-algorithm-35013.patch	4,01 ko	24 juillet 2019	Benjamin Dauvergne
0001-template-filter-cleaning-numbers-35013.patch	2,12 ko	24 juillet 2019	Benjamin Dauvergne
0001-template-filter-for-Luhn-algorithm-35013.patch	4,89 ko	24 juillet 2019	Benjamin Dauvergne
0001-fields-add-Luhn-algorithm-to-string-field-validation.patch	4,77 ko	10 août 2019	Benjamin Dauvergne
0001-fields-add-Luhn-algorithm-to-string-field-validation.patch	5,67 ko	13 août 2019	Benjamin Dauvergne