# w.c.s. - Development #31268

## Templatetag pour générer un jeton aléatoire

11 mars 2019 12:34 - Benjamin Dauvergne

| | | | |
|---|---|---|---|
| **Statut:** | Fermé | **Début:** | 11 mars 2019 |
| **Priorité:** | Normal | **Echéance:** | |
| **Assigné à:** | Benjamin Dauvergne | **% réalisé:** | 0% |
| **Catégorie:** | | **Temps estimé:** | 0:00 heure |
| **Version cible:** | | | |
| **Patch proposed:** | Oui | **Planning:** | |

**Description**

Syntaxes proposés:

```
{% random-decimal 4 %} -> 7585
{% random-alphanum 5 %} -> 17UX4 (majuscules + chiffres - {1, I, 0, O})
{% random-decimal 4 seed1 seed2 as jeton %} -> code secret sur 4 chiffres lié au hash de seed1 see
d2 stocké dans variable jeton
```

Dans le cas sans graîne on doit stocker par ailleurs ce code, dans le cas avec
graîne il faut récupérer la graîne avant de regénérer et valider lors du test.

### Révisions associées

**Révision bbeb4033 - 05 avril 2019 13:54 - Benjamin Dauvergne**

templatetags: add tags for token generation/validation (#31268)

- add {% token_decimal n } ~~-> n digits random token~~
add { token_alphanum n %} ~~-> n digits/uppercase-letters (without 0,1,I and O) random token~~
token1|token_check:token2 -> verify token1 is equal to token2
insensitive to case and prefix/suffix spaces.

### Historique

**#1 - 11 mars 2019 12:52 - Frédéric Péters**

> {% random-decimal 4 %} -> 7585

Pas sûr que les tirets passent; mais avant ça, c'est pour quoi faire ?

**#2 - 11 mars 2019 13:03 - Benjamin Dauvergne**

C'est pour mettre en place un formulaire d'appairage complètement géré par nous coté CD06 avec envoie du code via mail/mobile venant du logiciel métier, il n'y a pas encore de ticket coté CD06 tout est passé par mail.

L'idée est de demander le numéro de dossier à la personne, de récupérer son profil depuis le logiciel métier, d'afficher :

```
Recevoir un code d'appairage par :
( ) SMS sur le 06-..-..-..-93
( ) mail sur g.......y@gmail.com
```

de générer un code via le templatetag, le stocker dans une donnée de traitement puis envoyer le code sur le dit numéro de mobile/ou l'email.

J'ai ajouté l'histoire des seeds parce qu'on pourrait aussi ne pas le stocker en faisant juste {% random-decimal 4 form_number dossier_data_mobile %}.

Idéalement il faudrait limiter le mombres de soumissions à 3 avant d'obliger à soumettre une nouvelle demande.

Une fois cette validation faite on crée une liaison sans validation coté logiciel métier, cette liaison devrait être temporaire (mais c'est un autre sujet qui concerne cette fois le connecteur).

**#3 - 11 mars 2019 14:52 - Benjamin Dauvergne**

*- Fichier 0001-templatetags-add-tags-for-token-generation-validatio.patch ajouté*

*- Statut changé de Nouveau à Solution proposée*

*- Patch proposed changé de Non à Oui*

#### #4 - 11 mars 2019 15:10 - Benjamin Dauvergne

J'ai changer les noms et ajouter un filtre :

- {% token_decimal 4 %} -> 1234
- {% token_alphanum 5 %} -> AB345
- {% token_decimal 4 seed1 seed2 %} -> 1234 déterministe par rapport à seed1 et seed2
- {% if token1|token_check:token2 %}ok{% endif %} vérifie que token1 == token2 en ignorant la casse (pour le cas alphanum) j'hésite à ajouter des .strip()

#### #5 - 11 mars 2019 15:13 - Benjamin Dauvergne

*- Fichier 0001-templatetags-add-tags-for-token-generation-validatio.patch ajouté*

#### #6 - 11 mars 2019 15:25 - Frédéric Péters

Vraisemblablement soucis de copié/collé,

```
+@register.simple_tag
+def standard_text(text_id):
+    return mark_safe(TextsDirectory.get_html_text(str(text_id)))
```

#### #7 - 11 mars 2019 15:44 - Thomas Noël

Je retirerais token_check, parce qu'un simple « token_saisi == token_enregistre » fera l'affaire et n'a pas besoin de documentation particulière ; s'il faut être plus souple sur la casse et les espaces cela pourra être « token_saisi|cut:" "|upper », au choix de la personne qui configure l'affaire.

#### #8 - 12 mars 2019 07:44 - Benjamin Dauvergne

Thomas Noël a écrit :

> Je retirerais token_check, parce qu'un simple « token_saisi == token_enregistre » fera l'affaire et n'a pas besoin de documentation particulière ; s'il faut être plus souple sur la casse et les espaces cela pourra être « token_saisi|cut:" "|upper », au choix de la personne qui configure l'affaire.

C'est typiquement le genre d'endroit/moment (ergonomie/sécu) où on n'a pas envie de réfléchir justement, IMHO (quitte à redonder un peu je trouve).

#### #9 - 12 mars 2019 07:47 - Benjamin Dauvergne

Aussi si vous avez des remarques plus sur le fond ça m'intéresse (genre si vous pensiez à d'autres façons d'assurer ce scénario).

#### #10 - 12 mars 2019 07:53 - Benjamin Dauvergne

*- Fichier 0001-templatetags-add-tags-for-token-generation-validatio.patch ajouté*

Sans le copié/collé inutile.

#### #12 - 12 mars 2019 09:00 - Frédéric Péters

> {% random-alphanum 5 %} -> 17UX4 (majuscules + chiffres - {1, I, 0, O})

Dans le patch il n'y a pas non plus XYZ ?

> random.SystemRandom()

À d'autres endroits on instancie une seule fois un objet depuis le random.SystemRandom().

Si l'histoire des seeds ne va pas être utilisée, je serais pour ne pas la mettre dans le code.

```
    if length > 100:
       return 'ERROR**TOKEN TOO LONG'
```

Je ne vois pas la motivation à gérer cette situation ainsi, soit accepter n'importe quoi, soit exploser davantage, que ça laisse une trace dans les erreurs logguées.

#### #14 - 12 mars 2019 10:40 - Benjamin Dauvergne

Frédéric Péters a écrit :

> {% random-alphanum 5 %} -> 17UX4 (majuscules + chiffres - {1, I, 0, O})

Dans le patch il n'y a pas non plus XYZ ?

Ok, problème de révision de l'alphabet.

> random.SystemRandom()

À d'autres endroits on instancie une seule fois un objet depuis le random.SystemRandom().

Si l'histoire des seeds ne va pas être utilisée, je serais pour ne pas la mettre dans le code.

[...]

Je suis un peu en avance de phase sur la réflexion de Mike donc je ne sais pas garantir que ce ne sera pas utilisé, je pensais que ça forcerait à avoir un code toujours valable tant que mail/mobile ne changent pas mais en fait en intégrant form_number aux graines ça rend le code unique par demande de raccordement.

> Je ne vois pas la motivation à gérer cette situation ainsi, soit accepter n'importe quoi, soit exploser davantage, que ça laisse une trace dans les erreurs logguées.

Ok.

#### #15 - 12 mars 2019 10:57 - Benjamin Dauvergne

*- Fichier 0001-templatetags-add-tags-for-token-generation-validatio.patch ajouté*

*- Fichier 0002-templatetags-add-token-generation-with-seeds-31268.patch ajouté*

- alphabet corrigé
- ValueError() utilisé si length > 37
- aspect seed séparé dans un deuxième commit.

#### #17 - 04 avril 2019 09:35 - Frédéric Péters

Je notais :

> À d'autres endroits on instancie une seule fois un objet depuis le random.SystemRandom().

C'est quelque chose que j'aimerais avoir ici aussi.

> 1. 128bits security is enough; for alphanum, 128 / log(32) = 36.9...

On comprends que ça explique le > 37 qui suit mais ça mériterait un peu de contexte avant pour dire de quelle sécurité on parle et pourquoi 128 bits est suffisant.  Aussi, la longueur devrait dépendre de la taille de l'alphabet, non ? (genre l'appel token_decimal demanderait une longueur supérieure).

À ce sujet, mettre une valeur par défaut aux paramètres length ?

#### #18 - 04 avril 2019 12:02 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Je notais :
>
> > À d'autres endroits on instancie une seule fois un objet depuis le random.SystemRandom().
>
> C'est quelque chose que j'aimerais avoir ici aussi.

Ok.

> 1. 128bits security is enough; for alphanum, 128 / log(32) = 36.9...

128bits d'entropie est la sécurité suffisante en général, pas besoin de plus; un attaque brute force est largement impossible, c'est totalement arbitraire mais souvent cité; mais j'ai viré la condition, si un client veut générer un token de 100 caractère tant pis pour lui.

> On comprends que ça explique le > 37 qui suit mais ça mériterait un peu de contexte avant pour dire de quelle sécurité on parle et pourquoi 128 bits est suffisant.  Aussi, la longueur devrait dépendre de la taille de l'alphabet, non ? (genre l'appel token_decimal demanderait une longueur supérieure).

Je vire tout ça c'est juste pas utile d'en discuter; et oui à difficulté équivalente decimal demande à être plus long, on l'expliquera éventuellement dans la doc.

> À ce sujet, mettre une valeur par défaut aux paramètres length ?

Ok j'ai mis 6 pour décimal (au pif ça correspond à une longueur fréquente pour des OTPs style 3D-Secure ou OATH) et 4 pour alphanum (ça correspond à peu près au même niveau de sécurité, log(10^6)/log(28) = 4.14).

Pour ne pas penser à tout ça on peut aussi prévoir d'envoyer un nouveau code à chaque essaie dans le workflow concerné.

#### #19 - 04 avril 2019 12:08 - Benjamin Dauvergne

*- Fichier 0001-templatetags-add-tags-for-token-generation-validatio.patch ajouté*

*- Fichier 0002-templatetags-add-token-generation-with-seeds-31268.patch ajouté*

Voilà, je ne demande à pouvoir commiter que le premier patch, le deuxième je le laisse en souvenir.

#### #20 - 04 avril 2019 14:27 - Frédéric Péters

Tu peux juste faire ''.join([r.choice(alphabet) for x in range(length)]

#### #21 - 04 avril 2019 18:04 - Benjamin Dauvergne

*- Fichier 0001-templatetags-add-tags-for-token-generation-validatio.patch ajouté*

*- Fichier 0002-templatetags-add-token-generation-with-seeds-31268.patch ajouté*

Voilà.

#### #22 - 05 avril 2019 13:46 - Frédéric Péters

*- Statut changé de Solution proposée à Solution validée*

oK 0001.

#### #23 - 05 avril 2019 13:55 - Benjamin Dauvergne

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

```
commit bbeb4033ba21fbfcdbdd640a2d4a345139f68297 (HEAD -> master, origin/master, origin/HEAD)
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Mon Mar 11 14:41:08 2019 +0100

    templatetags: add tags for token generation/validation (#31268)

    - add {% token_decimal n %} -> n digits random token
    - add {% token_alphanum n %} -> n digits/uppercase-letters (without
      0,1,I and O) random token
    - token1|token_check:token2 -> verify token1 is equal to token2
      insensitive to case and prefix/suffix spaces.
```

#### #24 - 05 avril 2019 18:16 - Frédéric Péters

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

### Fichiers

| | | | |
|---|---|---|---|
| 0001-templatetags-add-tags-for-token-generation-validatio.patch | 4,34 ko | 11 mars 2019 | Benjamin Dauvergne |
| 0001-templatetags-add-tags-for-token-generation-validatio.patch | 4,73 ko | 11 mars 2019 | Benjamin Dauvergne |
| 0001-templatetags-add-tags-for-token-generation-validatio.patch | 4,61 ko | 12 mars 2019 | Benjamin Dauvergne |
| 0001-templatetags-add-tags-for-token-generation-validatio.patch | 3,77 ko | 12 mars 2019 | Benjamin Dauvergne |

| | | | |
|---|---|---|---|
| 0002-templatetags-add-token-generation-with-seeds-31268.patch | 3,47 ko | 12 mars 2019 | Benjamin Dauvergne |
| 0001-templatetags-add-tags-for-token-generation-validatio.patch | 4,13 ko | 04 avril 2019 | Benjamin Dauvergne |
| 0002-templatetags-add-token-generation-with-seeds-31268.patch | 3,85 ko | 04 avril 2019 | Benjamin Dauvergne |
| 0001-templatetags-add-tags-for-token-generation-validatio.patch | 4,07 ko | 04 avril 2019 | Benjamin Dauvergne |
| 0002-templatetags-add-token-generation-with-seeds-31268.patch | 3,85 ko | 04 avril 2019 | Benjamin Dauvergne |