

## w.c.s. - Development #29337

### calculs sur les dates dans les gabarits Django

24 décembre 2018 16:08 - Thomas Noël

<b>Statut:</b> Fermé	<b>Début:</b> 24 décembre 2018
<b>Priorité:</b> Normal	<b>Echéance:</b>
<b>Assigné à:</b>	<b>% réalisé:</b> 0%
<b>Catégorie:</b>	<b>Temps estimé:</b> 0:00 heure
<b>Version cible:</b>	<b>Planning:</b>
<b>Patch proposed:</b> Oui	
<b>Description</b>	
Ajouter des filtres tels que add/sub/mul/div pour faire des calculs, par exemple 5:add:3 renvoie 8.	
Un objectif important est d'obtenir le fonctionnement sur les dates, form_var_x add:4 ajouterait 4 jours.	
<b>Demandes liées:</b>	
Lié à w.c.s. - Development #57982: Ajouter les filtres add_months et add_years	<b>Nouveau</b> <b>19 octobre 2021</b>

#### Révisions associées

##### Révision 853c0e8e - 28 janvier 2019 15:19 - Thomas Noël

templatetags: add add\_days and add\_hours filters (#29337)

#### Historique

##### #1 - 24 décembre 2018 16:16 - Thomas Noël

- Fichier 0001-templatetags-add-maths-filters-29337.patch ajouté
- Statut changé de Nouveau à En cours
- Patch proposed changé de Non à Oui

Ci-joint un patch dont je ne suis pas pleinement satisfait. Je voulais être un peu magique et permettre que « form\_var\_date|add:2 » renvoie bien la date + 2 jours. L'usage bien sur est pour une comparaison, de type « form\_var\_date|add:2 < now ». Mais pépin : Python ne compare pas date et datetime, il faut donc caster pour être sûr de son coup, genre « form\_var\_date|add:2 < now|datetime » et l'affaire devient vite incompréhensible au néophyte (voir la fin des tests) et également à l'expert qui ne se souviendra par toujours de ce que fait la magie...

Et donc, je suis sur le point de revoir ce patch, supprimer la partie "guess\_date", et donc imposer les cast lors des comparaisons de date, qui devront être explicites comme « form\_var\_date|date|add:2 < now|date » ou « form\_var\_dateheure|datetime|add:"36h" < now|datetime » (ça semble plus lourd, mais ça ne me semble pas inexplicable).

Je veux bien un petit avis sur le sujet, avant de continuer à creuser, au cas où une bonne idée m'échapperait.

##### #2 - 24 décembre 2018 19:40 - Thomas Noël

- Fichier 0001-templatetags-add-maths-filters-29337.patch ajouté
- Statut changé de En cours à Solution proposée

Une version avec moins de magie, qui impose de convertir les valeurs en date ou datetime avant de pouvoir y faire un add. On se retrouve ainsi à avoir systématiquement form\_var\_x|date|add:"2" ou form\_var\_x|datetime|add:"30m". A cet instant de ma réflexion, peut-être laisser add pour les nombres seulement, et créer des filtres date\_add et datetime\_add spécialisés.

##### #3 - 25 décembre 2018 06:43 - Pierre Cros

Quand j'essaie de lister mes usages principaux (quasi exclusifs en fait), ça donne :

- Envoyer un mail x jours avant une date donnée (date de rendez-vous)
- Comparer deux dates
- Ajouter/soustraire des jours à une date

Et du coup c'est ça qui devrait être simple/magique. Alors je mets ce que je trouve simple comme syntaxe pour ces 3 exemples, sans présager de la faisabilité (parce que j'ai conscience de mon ignorance).

```
* form_var_date_envoi == form_var_date_rendez_vous|sub:2
* form_var_date1 == form_var_date2 ou form_var_date1 > form_var_date2
* form_var_date == form_var_date|add:10
```

Et encore plus simple et sans doute encore moins possible, se serait un filtre unique calc ou calc\_date, me permettant de faire |calc\_date:"+1" ou |calc\_date:"-2"

#### #4 - 25 décembre 2018 13:47 - Thomas Noël

Pierre Cros a écrit :

- form\_var\_date\_envoi form\_var\_date\_rendez\_vous|sub:2
- form\_var\_date1 form\_var\_date2 ou form\_var\_date1 > form\_var\_date2
- form\_var\_date == form\_var\_date|add:10

Malheureusement les dates sont stockées sous forme de chaîne de caractères dont le format n'est pas connu, voire d'autres formats, et ça peut être "21/12/2017" ou "2017-12-21" ou "2017-12-21 10:30"... du coup impossible de faire des comparaisons ou des traitements. Il faut passer par de la conversion en objet de type "date", et donc ça donnerait :

```
* form_var_date_envoi|date == form_var_date_rendez_vous|date|sub:2
* form_var_date1|date == form_var_date2|date ou form_var_date1|date > form_var_date2|date
* form_var_date|date == form_var_date|date|add:10
```

Ce qui me semble presque explicable, j'imagine quelque chose dans la doc comme : « Quand vous devez comparer des dates ou faire des calculs sur les dates, vous devez d'abord convertir les données en "objet date". Pour cela, après le nom de la variable, ajoutez le filtre "|date". » Et avec des exemples ensuite.

Par ailleurs dans le patch ci-dessus, add permet d'ajouter des minutes ou des heures à des objets de type "datetime", plus précis que la date. Par exemple pour envoyer un mail 24h avant un rendez-vous : « form\_var\_rdv|datetime|sub:"24h" » qui sera à comparer avec « now|datetime ».

#### #5 - 26 décembre 2018 09:21 - Frédéric Péters

En vrac, dans différentes directions,

- si les types doivent être explicites, il faut que la notion soit acceptée/assimilée, pour ensuite être bien exposée dans la documentation;
- si le seul truc qui empêche de commencer l'opération add par un parse\_date est l'argument form\_var\_date|add:2 < now, non comparaison date/datetime, je dirais que ça se joue là en deux parties : 1/ la date c'est "today", pas "now", 2/ peut-on exploiter LazyFieldVarDate, pour se donner de la marge sur les opérateurs ?
- perso je trouverais plus simple d'avoir add\_years/add\_days/add\_minutes que add: + quantité et unité, plus simple d'expliquer qu'à une date tu peux ajouter des années (add\_years), des jours (add\_days), des minutes (add\_minutes), que d'avoir un gros pavé décrivant tout ce qui peut être tapé dans "add", avec des risques d'explications impossibles parce que "magie".
  - ça permet aussi d'avoir des enchainements, façon form\_date\_plop|add\_days:2|add\_minutes:60, alors que si on part sur un add magique, je crains qu'on tende vers form\_var\_plop|add:"2d60m".

#### #6 - 26 décembre 2018 10:22 - Frédéric Péters

- *Sujet changé de calculs dans les gabarits Django, notamment pour les dates à calculs dans les gabarits Django, notamment pour les dates*

#### #7 - 26 décembre 2018 15:51 - Pierre Cros

J'ai édité mon message parce que les

==

avaient été bouffés.

Pour moi l'ajout d'heures ou de minutes c'est un truc tout à fait accessoire, je ne l'ai jamais utilisé. Alors ça ne me gênerait pas que ça reste compliqué (pour les gens qui aiment bien se compliquer la vie pour rien), tandis que les opérations sur les jours resteraient simples.

#### #8 - 07 janvier 2019 11:21 - Benjamin Dauvergne

Je trouve que spécifier l'unité dans add me semble plus clair (et ça évite un cast explicite que les gens oublieront), je met mon +1 sur |add\_day(s):x.

#### #9 - 18 janvier 2019 14:40 - Frédéric Péters

- *Sujet changé de calculs dans les gabarits Django, notamment pour les dates à calculs dans les gabarits Django (uniquement les dates ici)*

Dans ce qui est noté aujourd'hui sur <https://doc-publik.entrouvert.com/admin-fonctionnel/parametrage-avance/expressions/> :

|days va permettre de soustraire ou d'ajouter des jours à une date. (...) form\_var\_date|parse\_date|days:-2 >= today|parse\_date

Ici, si je lis bien, on va arriver à faire form\_var\_date|add\_days:-2 >= today ; correct ?

(penser à ajouter des tests de comparaison dans le commit)

## #10 - 18 janvier 2019 16:03 - Thomas Noël

- Fichier 0001-templatetags-add-add\_days-and-add\_hours-filters-2940.patch ajouté

- Sujet changé de calculs dans les gabarits Django (uniquement les dates ici) à calculs sur les dates dans les gabarits Django

Frédéric Péters a écrit :

Dans ce qui est noté aujourd'hui sur <https://doc-publik.entrouvert.com/admin-fonctionnel/parametrage-avance/expressions/> :

```
|days va permettre de soustraire ou d'ajouter des jours à une date. (...) form_var_date|parse_date|days:-2 >= today|parse_date
```

Ici, si je lis bien, on va arriver à faire `form_var_date|add_days:-2 >= today` ; correct ?

Oui, ce que je propose dans la branche <http://git.entrouvert.org/wcs.git?h=wip/29406-lazy-date-system> dont j'extrais le patch ici. Il ajoute juste `add_days` et `add_hours`, disons que c'est une phase 1 de l'affaire. Et ça permet :

```
form_var_date|add_days:-2 >= today
form_var_date|add_days:form_option_delai >= today
form_var_date >= today|add_days:2
form_var_date >= form_var_autredate
```

J'ai ajouté des tests sur ça, et d'autres.

## #11 - 22 janvier 2019 10:31 - Frédéric Péters

À relire, l'import de copy est inutile.

## #12 - 25 janvier 2019 17:31 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

## #13 - 28 janvier 2019 22:51 - Thomas Noël

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

Après suppression de l'import inutile de copy

```
commit 853c0e8e2b6365e8560ee0576be7c6066d15e668
Author: Thomas NOEL <tnoel@entrouvert.com>
Date: Fri Jan 18 09:21:09 2019 +0100
```

```
templatetags: add add_days and add_hours filters (#29337)
```

## #14 - 28 janvier 2019 23:15 - Frédéric Péters

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

## #15 - 19 octobre 2021 15:05 - Mikaël Ates (de retour le 29 avril)

- Lié à Development #57982: Ajouter les filtres `add_months` et `add_years` ajouté

## Fichiers

0001-templatetags-add-maths-filters-29337.patch	10,9 ko	24 décembre 2018	Thomas Noël
0001-templatetags-add-maths-filters-29337.patch	11,8 ko	24 décembre 2018	Thomas Noël
0001-templatetags-add-add_days-and-add_hours-filters-2940.patch	8,28 ko	18 janvier 2019	Thomas Noël