

## w.c.s. - Development #71430

### Saut automatique : possibilité de préciser une date

17 novembre 2022 17:34 - Pierre Ducroquet

<b>Statut:</b>	Rejeté	<b>Début:</b>	17 novembre 2022
<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>	Non
<b>Patch proposed:</b>	Non		
<b>Description</b>			
On observe a minima un cas où notre mécanique d'expiration est insuffisante pour répondre complètement à un besoin et est complétée par une condition pour spécifier une date. On est donc obligés de charger et d'évaluer inutilement l'ensemble des données, tout en ayant au final un système moins clair pour l'utilisateur.			
<b>Demandes liées:</b>			
Lié à w.c.s. - Development #76047: Discussion: mettre en cache en base la dat...		<b>Fermé</b>	<b>30 mars 2023</b>

### Historique

#### #2 - 17 novembre 2022 17:46 - Frédéric Péters

"une date" : entendu vraiment comme une date, c'est-à-dire un workflow où il serait noté 17/11/2022 ?

(à suivre les liens, oui, là il y a vraiment un workflow qui met en dur "2023-06-30").

Ça m'a l'air globalement farfelu, à un point où en fait ça ne fonctionnera quasi jamais : la date ne sera généralement pas mise ainsi dans le workflow mais dans une option et donc il y aura gabarit à évaluer, et donc fin du game.

#### #3 - 17 novembre 2022 18:53 - Pierre Ducroquet

Je comprends l'argument que ça semble farfelu, mais Brice semblait dire qu'il avait déjà eu le besoin, et si on a au moins cette possibilité affichée dans l'interface, même si tout le monde ne l'utilisait pas, le peu de cas sera toujours bénéfique par rapport au volume de calculs inutiles dans le cas contraire.

#### #4 - 17 novembre 2022 18:59 - Frédéric Péters

- Statut changé de Nouveau à Information nécessaire

- Assigné à mis à Brice Mallet

Je suis pour renforcer/automatiser la détection des situations problématiques mais vraiment je ne crois pas à l'option très limitée/spécifique ici.

mais Brice semblait dire qu'il avait déjà eu le besoin

(détails nécessaire)

#### #5 - 17 novembre 2022 19:24 - Pierre Cros

Ce qui permettrait de clarifier les expirations m'intéresse (même si j'ai bien compris que le but premier du ticket c'est de diminuer le volume de calcul).

Cela étant, contrairement à Brice, je n'ai jamais eu besoin de mettre une date fixe dans un saut auto

#### #6 - 18 novembre 2022 04:35 - Thomas Noël

Pierre Cros a écrit :

Cela étant, contrairement à Brice, je n'ai jamais eu besoin de mettre une date fixe dans un saut auto

Et pour ma part je rejoins finalement Frédéric, ça ne sera jamais le cas en réalité, on aura plutôt quelque chose comme

```
{{ form_option_date_de_bascule }}
```

et jamais une date fixe perdue au fin fond de la configuration d'un saut dans un workflow.

#### #7 - 18 novembre 2022 08:41 - Benjamin Dauvergne

Thomas Noël a écrit :

Et pour ma part je rejoins finalement Frédéric, ça ne sera jamais le cas en réalité, on aura plutôt quelque chose comme

```
{{ form_option_date_de_bascule }}
```

et jamais une date fixe perdue au fin fond de la configuration d'un saut dans un workflow.

Ça change quoi que la date soit fixe ou variable ? Si l'objectif c'est d'avoir une colonne "next\_expiration" et y mettre une date pour savoir quand réveiller la demande plutôt que de la tester à chaque réveil du cron, c'est pareil (maintenant je m'avance peut-être un peu sur l'implémentation technique envisagée).

#### #8 - 18 novembre 2022 08:52 - Frédéric Péters

Ça change quoi que la date soit fixe ou variable ?

Si c'est un gabarit ça veut dire qu'il faut le calculer, et que le résultat peut varier selon les demandes, qu'il faut donc passer sur toutes les demandes/fiches.

#### #9 - 18 novembre 2022 09:47 - Benjamin Dauvergne

Frédéric Péters a écrit :

Ça change quoi que la date soit fixe ou variable ?

Si c'est un gabarit ça veut dire qu'il faut le calculer, et que le résultat peut varier selon les demandes, qu'il faut donc passer sur toutes les demandes/fiches.

Pour une demande donnée le résultat ne peut pas varier sauf si il dépend de quelque chose qui n'est pas dans le formdata, c'est peut-être ça qu'il faudrait pouvoir détecter et réévaluer à chaque formdata.store(), une date du genre `{{ form_last_update_time|add_days:10 }}` on peut en stocker la valeur dans une colonne expiration si c'est recalculé à chaque formdata.store(), on est sûr que ça va marcher (et sur une modification du gabarit dans le workflow il faut repasser sur toutes les demandes dans le statut correspondant mais ce n'est pas pire que maintenant).

Je ne dis pas que c'est simple de ne plus tester systématiquement les sauts automatiques avec condition dans tous les cas. Mais on doit pouvoir progressivement extraire les cas "simples" en étant prudent : les conditions et expirations qui ne dépendent que du dernier statut et de l'état courant de la demande (typiquement la présence d'un webservice.\* rend ça faux), pour une condition simple, si elle est fausse maintenant elle le sera toujours dans le futur si la demande n'est pas modifiée entre temps. Et pour une date d'expiration elle ne changera pas, elle peut donc être stockée et servir de filtre.

Ça marcherait d'ailleurs bien avec les triggers aussi, sur un trigger la demande est modifiée et le statut de toutes les conditions réévaluées (en posant un `formdata.expiration = now()` pour que la demande soit considérée à la prochain boucle par exemple).

#### #10 - 19 novembre 2022 11:13 - Brice Mallet

- Statut changé de Information nécessaire à Nouveau

- Assigné à Brice Mallet supprimé

Et pour ma part je rejoins finalement Frédéric, ça ne sera jamais le cas en réalité, on aura plutôt quelque chose comme `{{ form_option_date_de_bascule }}`

Comparaison à une date, qu'elle soit en dur ou dans une variable de WF (il faudra encourager la variable, certainement pour des questions de maintenance) est le cas : année scolaire, date de commission, date max pour déposer une demande de sub...

#### #11 - 19 novembre 2022 11:31 - Frédéric Péters

il faudra encourager la variable

C'est mon propos : on n'encouragera pas de date fixe, on sera sur un gabarit, et ça ne fait au final donc pas de différence (de calcul) par rapport à la condition django.

Reste ce que Benjamin écrit, « on doit pouvoir progressivement extraire les cas "simples" en étant prudent » mais ça peut totalement passer sans ajouter une "possibilité de préciser une date". Et ça n'est donc pas ce ticket.

#### #12 - 22 novembre 2022 14:34 - Frédéric Péters

- Statut changé de Nouveau à Rejeté

Et donc, non.

#### #13 - 23 novembre 2022 17:03 - Benjamin Dauvergne

« mais ça peut totalement passer sans ajouter une possibilité de préciser une date". »

Pas sûr.

Parce qu'autant c'est simple de dire, dans le cas :

- Quand: Toutes les 20 minutes
- Condition: form\_var\_truc "Oui"  
que si la condition est fausse est fausse maintenant elle le sera encore dans 20 minutes et donc pas la peine de programmer un test de la demande dans 20 minutes.

Pareil pour :

- Quand: Toutes les 20 minutes
- Condition: webservice.truc.machin "Oui"

De la présence de webservice on peut déduire qu'on ne peut rien savoir et donc on se tapera un test toutes les 20 minutes.

Autant analyser ça :

- Quand: Toutes les 20 minutes
- Condition: form\_var\_truc == "Oui" and form\_var\_date|date|add\_days:30 <= today|date

pour en déduire qu'il faut programmer le test de la demande dans 30j c'est une analyse plus compliqué

Alors si on pouvait directement le remplacer par:

- Quand: {{ form\_var\_date|date|add\_days:30 }}
- Condition: form\_var\_truc == "Oui"

je crois qu'on gagnerait un cas simple facilement.

#### #14 - 30 mars 2023 20:04 - Frédéric Péters

- Lié à [Development #76047](#): Discussion: mettre en cache en base la date du prochain réveil pour les actions globales déclenchés par un réveil ou pour les sauts sur expiration ajouté