

w.c.s. - Development #56847

Action de Wf externe : pouvoir spécifier une liste d'identifiants sur lequel l'action doit agir à partir d'une requête

10 septembre 2021 11:11 - Stéphane Laget

Statut:	Fermé	Début:	10 septembre 2021
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Parfois on souhaite déclencher un wf externe sur une liste de fiches précise.			
L'idée serait de pouvoir récupérer la liste des fiches liées filtrées, et spécifier que l'action de wf externe ne doit pouvoir s'exécuter que sur ces fiches.			
avec une requête type <code>{{ cards objects:"XXXXX" filter_by_user:form_user getlist:"form_internal_id }}</code>			
Demandes liées:			
Duplique w.c.s. - Development #51240: Pouvoir déclencher un wf externe sur pl...		Fermé	18 février 2021

Révisions associées

Révision 5f3ada3b - 02 novembre 2021 16:48 - Frédéric Péters

workflows: allow queryset as id for external workflow action (#56847)

Historique

#1 - 10 septembre 2021 11:11 - Stéphane Laget

- *Sujet changé de Action de Wf externe : pouvoir spécifier une liste d'identifiants sur lequel l'action doit agir à partir d'une requête à Action de Wf externe : pouvoir spécifier une liste d'identifiants sur lequel l'action doit agir à partir d'une requête*

#2 - 10 septembre 2021 12:36 - Thomas Noël

Idée : pour l'instant pour attraper le numéro de la cible on fait un

```
target_id = self.compute(self.target_id, formdata=formdata, status_item=self)
```

il faudrait voir si en passant en mode `allow_complex=True` ce qu'on pourrait récupérer d'une requête `{{ cards|objects:"..."|filter_... }}`, soit directement une fiche/demande, soit carrément une queryset, et dans ce cas jouer un itérateur, etc.

Ceci étant, ça sent quand même le danger : une erreur dans le filtre de requête et boom! toutes les fiches sont jouée... et aussi un risque de timeout/crash si le nombre de fiches ciblées est trop important. En terme d'usage on ouvre quand même pas mal de bogues potentiels.

#3 - 20 septembre 2021 10:13 - Benjamin Dauvergne

Et sinon sélectionner une vue dans une liste des vues disponibles ? On laisse la définition de la requête dans la définition de la vue, ça reste raccord avec les usages introduits dernièrement.

Par contre pour le côté noob proof je ne pense pas que quoi que ce soit entièrement possible, on peut juste s'arranger pour que ce soit fait en tâche de fond via uwsgi, et empêcher que ça s'emballe. On pourrait ainsi stocker la liste des id et le trigger à levé dans une table "FormJob(formdef_id, formdata_id, trigger_name, state)" et on laisse une tâche de fond faire tout ça, si on provoque le même trigger alors que le premier est toujours dans le statut "todo", on ne fait rien (donc pas possible de pousser des jobs plus vites qu'on ne les traite), tant qu'un signal est en cours, le provoquer deux fois ne fait rien de plus.

#4 - 20 septembre 2021 11:57 - Thomas Noël

Benjamin Dauvergne a écrit :

Et sinon sélectionner une vue dans une liste des vues disponibles ? On laisse la définition de la requête dans la définition de la vue, ça reste raccord avec les usages introduits dernièrement.

Oui, aussi. Ceci étant dans l'exemple cité « `filter_by_user:form_user` » je ne vois pas comment on va pouvoir exprimer ça clairement dans une vue. « La liste des fiches dont l'utilisateur est celui de la demande qui exécute le workflow externe » : pas simple. (Mais peut-être que ça existe déjà, je ne suis pas assez à jour sur les paramètres possibles dans les vues)

Par contre pour le coté noob proof je ne pense pas que quoi que ce soit entièrement possible, on peut juste s'arranger pour que ce soit fait en tâche de fond via uwsgi, et empêcher que ça s'emballe. On pourrait ainsi stocker la liste des id et le trigger à levé dans une table "FormJob(formdef_id, formdata_id, trigger_name, state)" et on laisse une tâche de fond faire tout ça, si on provoque le même trigger alors que le premier est toujours dans le statut "todo", on ne fait rien (donc pas possible de pousser des jobs plus vites qu'on ne les traite), tant qu'un signal est en cours, le provoquer deux fois ne fait rien de plus.

Je retiens de cela et la discussion qu'on vient d'avoir en réunion commune que :

- sur une action de workflow externe déclenchée sur un ensemble de fiches (vues ou requête) : on lance ça en asynchrone
- pour ne pas que tout soit complètement transparent-magique, et pour aider au support/suivi, il serait très utile que le niveau d'exécution du job soit visible au moins dans l'inspect (ça pourrait peut-être se faire via l'enregistrer du numéro de job dans un log ActionsTracingEvolutionPart ?)

#5 - 20 septembre 2021 12:25 - Frédéric Péters

sur une action de workflow externe déclenchée sur un ensemble de fiches (vues ou requête) : on lance ça en asynchrone

Ma position c'est rien d'automatique ou de paramétrable ici, juste instruction donnée aux personnes qui créent des workflows que s'il y a une action qui va tourner sur 1000 demandes, il faut qu'elle soit dans un statut dans lequel on arrive via une expiration, pour que ça soit joué en cron.

pour ne pas que tout soit complètement transparent-magique, et pour aider au support/suivi, il serait très utile que le niveau d'exécution du job soit visible au moins dans l'inspect (ça pourrait peut-être se faire via l'enregistrer du numéro de job dans un log ActionsTracingEvolutionPart ?)

Et donc pour moi on n'a pas de job (au sens AfterJob). Mais on peut quand il y a exécution sur un queryset créer un WhateverEvolutionPart avec dedans index et total.

#6 - 20 septembre 2021 12:36 - Thomas Noël

Frédéric Péters a écrit :

sur une action de workflow externe déclenchée sur un ensemble de fiches (vues ou requête) : on lance ça en asynchrone

Ma position c'est rien d'automatique ou de paramétrable ici, juste instruction donnée aux personnes qui créent des workflows que s'il y a une action qui va tourner sur 1000 demandes, il faut qu'elle soit dans un statut dans lequel on arrive via une expiration, pour que ça soit joué en cron.

Quand bien même, sur 50 000 demandes on est sûr que ça va se faire en moins de 4 ou 5 secondes (ie avant que l'agent pense que tout est planté) ? J'ai posé la question à Stéphane et oui, on est dans ces volumétries. Sans asynchrone j'ai du mal à imaginer la chose.

#7 - 20 septembre 2021 12:49 - Frédéric Péters

Dans mon déroulé il y a un bouton "Programmer la diffusion", l'agent clique dessus, ça met dans un statut qui affiche un message et qui a un saut expiration dans 20 minutes (ou autre, genre pour exécution la nuit). Ça prend moins de 5 secondes, bien sûr. Plus tard le cron arrive et il prend le temps qu'il lui faut.

#8 - 20 septembre 2021 13:50 - Thomas Noël

Frédéric Péters a écrit :

Dans mon déroulé il y a un bouton "Programmer la diffusion", l'agent clique dessus, ça met dans un statut qui affiche un message et qui a un saut expiration dans 20 minutes (ou autre, genre pour exécution la nuit). Ça prend moins de 5 secondes, bien sûr. Plus tard le cron arrive et il prend le temps qu'il lui faut.

Ah, compris, le saut à expiration est dans le workflow qui demande l'action, et pas dans le workflow externe qui l'exécute. Ça me va sur ça.

Reste la problématique de "voir que l'action workflow externe est en cours", comment imaginerais-tu la chose ? Il faudrait logger dans le formdata qu'on est en train de faire une action...?

#9 - 20 septembre 2021 14:10 - Frédéric Péters

(...) l'agent clique dessus, ça met dans un statut qui affiche un message et qui a un saut expiration dans 20 minutes

et là donc après ~20 minutes il y a un statut avec l'action "workflow externe" configurée avec une liste d'id; lors de l'exécution on ajoute le WhateverEvolutionPart et on l'actualise au fur et à mesure, façon

```

--- a/wcs/wf/external_workflow.py
+++ b/wcs/wf/external_workflow.py
@@ -221,6 +221,7 @@ class ExternalWorkflowGlobalAction(WorkflowStatusItem):

    caller_source = CallerSource(formdata)

+    # ici on ajoute un WhateverEvolutionPart dans l'historique
    formdata.store()
    for target_data in self.iter_target_datas(formdata, objectdef):
        with get_publisher().substitutions.temporary_feed(target_data):
@@ -230,6 +231,9 @@ class ExternalWorkflowGlobalAction(WorkflowStatusItem):
        get_publisher().substitutions.feed(target_data)
        get_publisher().substitutions.feed(caller_source)
        perform_items(trigger.parent.items, target_data)
+    # ici on met à jour l'index dans le WhateverEvolutionPart
+    ...
+    formdata.store()

```

Et ce WhateverEvolutionPart, il affiche l'info de progression si jamais un agent va sur la demande alors que l'action est en cours.

#10 - 16 octobre 2021 18:06 - Frédéric Péters

- Duplique Development #51240: Pouvoir déclencher un wf externe sur plusieurs fiches existantes sur la base d'un ciblage manuel ajouté

#11 - 16 octobre 2021 18:38 - Frédéric Péters

- Assigné à mis à Frédéric Péters

#12 - 29 octobre 2021 15:28 - Frédéric Péters

- Fichier 0001-workflows-allow-queryset-as-id-for-external-workflow.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Voilà c'est un peu plus de code que ce qui était noté ci-dessus mais ça reste l'idée.

#13 - 29 octobre 2021 19:19 - Thomas Noël

Sur les messages de mismatch, au lieu de "X vs Y" on gagnera un peu de temps avec un truc genre "type X attendu, mais type Y reçu" (genre "expect X, got y").

C'est moi qui rate un truc, ou les ManyExternalCallsPart vont s'afficher à tout le monde pendant leur exécution ? Autrement dit, il faudra avoir l'action dans un statut caché si nécessaire ? (ce qui n'est pas grave, je demande juste).

Pendant qu'on y est, et s'il pleut, un truc sympa serait que ManyExternalCallsPart contienne l'objectdef et la liste des 10 derniers target_data sur lesquels l'action a été lancée. J'ai dans l'idée que ça sera bien utile pour le support. On peut même imaginer que cette trace sera utile même s'il y a un seul élément cible -- donc avoir systématiquement un ExternalCallsPart sur toute action de workflow externe...?

#14 - 02 novembre 2021 13:04 - Frédéric Péters

- Fichier 0001-workflows-allow-queryset-as-id-for-external-workflow.patch ajouté

Message modifié et pour l'affichage c'est désormais limité au backoffice (j'ignorais la question en imaginant les usages réels sur des fiches).

Et enregistrement de toutes les demandes/fiches traitées, dès qu'il y en a une.

#15 - 02 novembre 2021 15:25 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Frédéric Péters a écrit :

Et enregistrement de toutes les demandes/fiches traitées

J'avais pensé limiter à 10 parce que je sais qu'un but de cette possibilité c'est de lancer une action sur des dizaines de milliers de fiches d'abonnement... On laisse filer ? (c'est vrai que ce n'est pas si lourd, quelques kilo-octets dans le statut ; je veux juste savoir si tu n'as pas oublié cet usage non-marginal).

En tout cas c'est ok pour moi ainsi.

#16 - 02 novembre 2021 15:29 - Frédéric Péters

Oui c'est pas lourd.

#17 - 02 novembre 2021 16:49 - Frédéric Péters

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

```
commit 5f3ada3bd858d533c34eb91830f4c8599548b45e
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Sat Oct 16 19:33:50 2021 +0200
```

```
workflows: allow queryset as id for external workflow action (#56847)
```

#18 - 02 novembre 2021 18:17 - Frédéric Péters

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

Fichiers

0001-workflows-allow-queryset-as-id-for-external-workflow.patch	13,7 ko	29 octobre 2021	Frédéric Péters
0001-workflows-allow-queryset-as-id-for-external-workflow.patch	14 ko	02 novembre 2021	Frédéric Péters