

w.c.s. - Development #36507

Clichés

28 septembre 2019 19:39 - Nicolas Roche

Statut:	Fermé	Début:	28 septembre 2019
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
<p>Créer 2 nouvelles vues pour gérer les clichés Les vues permettent de lister les clichés, d'en créer un nouveau, de les supprimer ou de les charger. Les clichés sont ordonnés par date (il n'y a pas d'arborescence à gérer).</p> <ul style="list-style-type: none">• backoffice/forms/XXX/snapshots• backoffice/workflows/XXX/snapshots <p>Stocker les clichés localement dans un nouveau répertoire, puis par type d'objet et enfin par l'identifiant de l'objet. Les clichés sont indépendants des objets (ils ne sont pas exportés avec).</p> <ul style="list-style-type: none">• tenant/snapshots/formdefs/XXX/• tenants/snapshots/workflows/XXX/ <p>Utiliser le mécanisme d'export et import/overwrite XML pour exploiter les clichés. Éventuellement en profiter pour ajouter le mécanisme de remplacement sur les workflows.</p> <ul style="list-style-type: none">• <code>admin/forms.py::overwrite_submit()</code>• <code>admin/workflows.py::import_submit()</code>			
Demandes liées:			
Lié à w.c.s. - Development #4960: Versionning sur formulaires et workflows		Fermé	13 juin 2014 07 septembre 2020
Lié à w.c.s. - Development #14138: "écraser avec un nouvel import" pour les w...		Nouveau	29 novembre 2016

Historique

#2 - 28 septembre 2019 20:18 - Nicolas Roche

- Description mis à jour

#3 - 28 septembre 2019 20:19 - Nicolas Roche

- Description mis à jour

#4 - 28 septembre 2019 20:24 - Nicolas Roche

- Lié à Development #4960: Versionning sur formulaires et workflows ajouté

#6 - 09 octobre 2019 10:39 - Nicolas Roche

En terme d'interface, clichés (versions) et historiques doivent être tout à fait liés :

- Il faudrait par exemple trouver le moyen d'indiquer les clichés qui "pointent" sur chaque ligne de logs.
(de façon analogue aux tags qui sont incrustés dans le 'git log')
- De l'autre côté (dans l'IHM des clichés), il faudrait par exemple trouver le moyen d'afficher (ou de basculer vers) les logs du cliché.

#7 - 09 octobre 2019 11:06 - Benjamin Dauvergne

Nicolas Roche a écrit :

En terme d'interface, clichés (versions) et historiques doivent être tout à fait liés :

- Il faudrait par exemple trouver le moyen d'indiquer les clichés qui "pointent" sur chaque ligne de logs.
(de façon analogue aux tags qui sont incrustés dans le 'git log')
- De l'autre côté (dans l'IHM des clichés), il faudrait par exemple trouver le moyen d'afficher (ou de basculer vers) les logs du cliché.

Me semble que le plus simple c'est de stocker les logs avec le cliché, ça rejoint le besoin d'exporter les logs dans les exports, il suffira de mettre une ligne dans le log chaque fois qu'un cliché est pris pour avoir des marqueurs pour faire correspondre logs et clichés. Quand on importe un cliché on écrase les logs et le pickle du formdef/workflow avec ce qu'il y a dans l'export.

#8 - 07 décembre 2019 14:39 - Nicolas Roche

- Lié à Development #14138: "écraser avec un nouvel import" pour les workflows ajoutés

#9 - 12 décembre 2019 16:42 - Nicolas Roche

- Fichier 0001-snapshots-store-snapshot-files-on-forms-update-36507.patch ajouté

- Fichier 0002-snapshots-inherit-and-adapt-forms-UI-to-display-sna.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Voici un mini POC (merci Thomas) pour les formulaires seulement :

- Les clichés sont de formulaires sauvegardés dans `/forms.snapshots/XX.date`
- Les clichés sont accessibles via l'URL `/backoffice/formsnaps/`

Est-ce que d'après vous je peux m'engager dans cette voie ?

#10 - 12 décembre 2019 17:34 - Nicolas Roche

- Fichier 0002-snapshots-inherit-and-adapt-forms-UI-to-display-sna.patch ajouté

(le second patch est incomplet)

#11 - 12 décembre 2019 22:53 - Thomas Noël

- Statut changé de Solution proposée à En cours

Un poil d'explication : j'ai proposé à Nicolas de creuser l'idée d'avoir des objets "FormDefSnapshot" qui sont en fait des FormDef, qui sont "readonly" (def store(): pass). Dès qu'un FormDef bouge (store) on enregistre un FormDefSnapshot.

On profitera alors de tout plein d'UI déjà prêtes pour afficher un snapshot de FormDef. Il faudra ensuite étudier des outils pour présenter la différence entre deux FormDef (y'a déjà beaucoup dans le système d'écrasement via import), présenter la liste des snapshot pour un formdef donné, nettoyer automatiquement les vieux snapshots.

L'idée serait à reprendre ensuite pour les workflows, sur le même principe (mais c'est bien de déjà "finir" les formdef, dans un premier temps, à mon avis).

Bref, si ce draft nous semble être une piste viable, ça devrait permettre à Nicolas de chiffrer sur une idée plus claire de ce qui doit être codé.

#12 - 13 décembre 2019 10:19 - Frédéric Péters

creuser l'idée d'avoir des objets "FormDefSnapshot" qui sont en fait des FormDef

À creuser... en l'état, 0001, "il y aura un objet par ancienne version, il sera créé automatiquement lorsqu'un formdef sera enregistré", peut tout à fait être un morceau d'un plan technique, mais oui, il faudrait creuser, avoir davantage pour voir vraiment la direction.

On est donc là sur un enregistrement automatique, mais dans une vue de l'historique, ce choix va créer une longue liste de versions connues uniquement par une date/heure. Est-il pertinent de pointer qu'il faudrait l'accompagner de quelques mots ? C'est-à-dire qu'une part du plan devrait être de passer sur les endroits où `.store()` est appelé pour y ajouter quelques mots de description, ex: "changement du workflow associé", ou "modification du champ Untel".

Et ce point qui semble un détail d'UI amène derrière quand même la question du stockage, parce qu'on va vouloir afficher la liste des versions, et peut-être qu'on peut se dire qu'ouvrir 50 pickles, c'est pas top, peut-être qu'on serait mieux à dire que les versions c'est postgresql uniquement, et travailler sur une table snapshots (object_type, object_id, timestamp, message, xml_export).

~~

0002, "mettre l'historique des formulaires au premier niveau de navigation", est une autre part du plan technique, que je ne pense pas appropriée; pour moi l'historique d'un formulaire doit être au niveau sous le formulaire, comme dans la description du ticket.

~~

(mais c'est bien de déjà "finir" les formdef, dans un premier temps, à mon avis).

Ça me va très bien de commencer par les formdefs.

#13 - 16 décembre 2019 10:13 - Thomas Noël

Frédéric Péters a écrit :

On est donc là sur un enregistrement automatique, mais dans une vue de l'historique, ce choix va créer une longue liste de versions connues uniquement par une date/heure. Est-il pertinent de pointer qu'il faudrait l'accompagner de quelques mots ? C'est-à-dire qu'une part du plan devrait être de passer sur les endroits où `.store()` est appelé pour y ajouter quelques mots de description, ex: "changement du workflow associé", ou "modification du champ Untel".

Il est prévu un algo de comparaison entre deux formdefs (reprendre/étendre celui qui gère l'écrasement par import). On peut imaginer que cet algo saura faire le calcul lors de l'enregistrement d'un snapshot (entre lui et son précédent) et que cette information sera enregistrée, et donc affichée dans le listing, oui. On aurait donc pour un formulaire donné une liste de ses clichés

- 10/10/2019 10h34 « ajout d'un champ bidule » par Tarte Enpion
- 10/10/2019 10h30 « ajout d'un champ chose » par Tarte Enpion
- 09/10/2019 16h01 « changement du workflow » par Tarte Enpion

Evidemment si on supprime un snapshot (par exemple le second) on perd l'info sur l'ajout du champ "chose". Il pourrait être recalculé... Ou alors on interdit de supprimer un snapshot, on retire juste, éventuellement, le XML stocké qui bouffe de la place (cf infra).

Et ce point qui semble un détail d'UI amène derrière quand même la question du stockage, parce qu'on va vouloir afficher la liste des versions, et peut-être qu'on peut se dire qu'ouvrir 50 pickles, c'est pas top, peut-être qu'on serait mieux à dire que les versions c'est postgresql uniquement, et travailler sur une table snapshots (object_type, object_id, timestamp, message, xml_export).

Yep. Et ajouter un chaînage "previous" qui pointe vers le snapshot précédent.

0002, "mettre l'historique des formulaires au premier niveau de navigation", est une autre part du plan technique, que je ne pense pas appropriée; pour moi l'historique d'un formulaire doit être au niveau sous le formulaire, comme dans la description du ticket.

L'idée de 0002 c'était pour moi de "convaincre" Nicolas qu'en ayant un objet FormDefSnapshot on gagnait déjà beaucoup d'UI. Pas grand chose de plus. Effectivement ça serait plus des pages /forms/58/snapshots/ (liste des clichés) et /forms/58/snapshots/32/ (affichage d'un cliché donné), etc.

Bref, il s'agissait ici d'aider Nicolas à chiffrer, et pour cela de creuser un peu une piste. Je pense que ce qu'il a fait, et ce début de discussion, suffit à montrer que c'est la voie à suivre.

On peut séparer le travail en plusieurs chantiers, à chiffrer :

- écrire un système de comparaison de deux formdef, `compare(formdef1, formdef2)` (3 jours)
 - qui renvoie une liste des modifs (modifs des options, champs ajoutés, champs supprimés, champs modifiés, ...) et un résumé en une ligne
- utilisation de ce système dans le processus d'overwrite (écraser avec un import) (2 jours)
- création d'un objet FormDefSnapshot dont le stockage se fait dans une table SQL (4 jours)
- interface de présentation des clichés liés à un formdef (1 jour)
- interface de présentation d'un cliché donné (2 jours)
- outil de restauration d'un cliché, reprenant ce qui a été dessiné dans "écraser avec un import" (exactement le même principe) (2 jours)

J'ai mis mon chiffrage "doigt mouillé" entre parenthèses, qui tape sur 14 jours... Tout est encore discutable.

#14 - 16 décembre 2019 10:20 - Frédéric Péters

Je ne voulais pas allonger ce ticket (à 14 commentaires j'ai déjà envie de le rejeter), inciter à faire un pad pour poser le fonctionnel et plan technique envisagé; Nicolas a commencé dans <https://pad.libre-entreprise.org/p/eo-snapshots>

#15 - 20 décembre 2019 12:02 - Nicolas Roche

- Statut changé de *En cours* à *Fermé*

repris par [#38663](#)

Fichiers

0001-snapshots-store-snapshot-files-on-forms-update-36507.patch	1,79 ko	12 décembre 2019	Nicolas Roche
0002-snapshots-inherite-and-adapt-forms-UI-to-display-sna.patch	2,03 ko	12 décembre 2019	Nicolas Roche
0002-snapshots-inherite-and-adapt-forms-UI-to-display-sna.patch	2,62 ko	12 décembre 2019	Nicolas Roche