

## Historisation et clichs

20 décembre 2019 11:54 - Nicolas Roche

<b>Statut:</b>	Rejeté	<b>Début:</b>	20 décembre 2019
<b>Priorité:</b>	Normal	<b>Echéance:</b>	07 septembre 2020
<b>Assigné à:</b>	Frédéric Péters	<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		

### Description

Le mot cliché employé ci-dessous fait référence à un instantané qui permet d'observer un objet.

L'historisation (Transformation de rites ou de mythes en récits historiques.) consiste ici en l'organisation chronologique de ces instantanés de sorte à former un historique.

L'objectif est de permettre en particulier de savoir qui a modifié quoi dans un formulaire, une fiche ou un workflows afin de faciliter la collaboration entre plusieurs intervenants amenés à travailler ensemble sur un même item. Afin de faciliter le retour à une version antérieure également.

Un cliché de l'item sera enregistré à chaque modification.

Une page "Historique" - accessible depuis la colonne de droite sur l'écran d'accueil d'un formulaire ou d'un workflow" - permettra de lister les différentes versions de l'item concerné.

Chaque ligne affichant une version contiendra :

- La date de la version
- Le nom de la personne ayant fait les modifications
- Un lien "voir"
- Un lien "restaurer"
- Un commentaire sur la nature de la dernière modification

Le lien "voir" permettra d'afficher la version concernée.

C'est à dire le workflow le formulaire ou la fiche en entier avec toutes les possibilités d'action en lecture seulement. En gros, tout ce qu'on peut voir via les pages présentées sous le menu studio, mais par contre pas de traitement ni de visualisation en ligne.

Le lien "restaurer" permettra de remplacer, après confirmation, l'item en cours par la version choisie. Cette restauration se basera sur les procédures actuelles d'import en place (remplacement) pour les formulaires et de duplication pour les workflows.

Par exemple, l'utilisateur sera averti lorsqu'une différence dans les champs du formulaire sera détectée. Et il devra explicitement valider l'opération une suppression de champs a été détectée.

Pour les workflows, l'utilisateur devra explicitement faire pointer les formulaires concernés vers le workflow restauré. Il s'agit ici en fait d'une duplication, plutôt que d'avoir à avertir systématiquement que quand on restaure un cliché, l'état courant sera perdu.

Les opérations de restauration étant par nature dangereuse pour les données des demandes en cours, je précise ici que l'utilisation principale des clichs est de pouvoir retrouver comment étaient configurés tels champs auparavant puis de reporter les corrections sur le formulaire ou workflow actuellement en cours d'utilisation.

Les clichs seront stockés en base de donnée.  
Ils ne seront pas persistants lors d'un ré-import.

Les vieux clichés seront détruits automatiquement afin de ne pas saturer la base de donnée.

#### Demandes liées:

Lié à w.c.s. - Development #4960: Versionning sur formulaires et workflows	Fermé	13 juin 2014	07 septembre 2020
Lié à Publik - Development #40832: Produire un journal de l'activité des util...	Nouveau	19 mars 2020	

## Historique

### #1 - 20 décembre 2019 12:00 - Nicolas Roche

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

### #2 - 20 décembre 2019 12:00 - Nicolas Roche

#### Décisions :

- ne pas ré-inventer la roue : on réutilise au maximum le code d'import des formulaire et des workflow
- le stockage se fait en base, avec un champs dédié blob pour l'objet lui même, exporté au format XML
- on enregistre un snapshot à chaque modification, ce qui finalement revient à gérer également l'équivalent des logs
- dans un premier temps, on ne s'occupe pas du diff

#### Plan:

1. gestion d'une tables qui liste les snapshots
  2. sauvegardes automatiques via création d'un dump XML puis insertion dans la table
  3. IHM de restauration des snapshots
  4. IHM d'énumération des versions via consultation de la table
  5. IHM de visualisation des snapshots copiée sur de l'IHM des items
- les item sont créés à la volée via import XML

#### Spécifications techniques :

1. gestion de la table snapshots inspirée de 'tracking\_code'  
(object\_type, object\_id, timestamp, message, xml\_export)

- sql.py::do\_tracking\_code\_table()
- sql.py::TrackingCode

2. sauvegardes automatiques

- Dès qu'un FormDef bouge (store) on enregistre un snapshot en base.
- Les snapshots sont "readonly" (def store(): pass).
- 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".

```
$ grep 'formdef.store()' wcs/admin/*.py | wc -l
15
$ grep 'workflow.store()' wcs/admin/*.py | wc -l
30
```

- ajouter 'message' en paramètre à store()

3. IHM de restauration d'une version (note de fred : c'est son propre sujet, indépendant d'une interface listant les versions, et ça serait à décliner formdef/workflow)

- récupération de l'item depuis la table puis import XML
- écrasement de l'item en recopiant tous les champs s'inspirant de

```
admin/forms.py::FormDefPage::overwrite()
admin/forms.py::FormDefPage::workflow_status_remapping()
```

- détection et avertissement quand il y a des champs en plus ou en moins sur les formulaires/fiches
- pour les workflows, on en crée un nouveau à côté avec le nom qui va bien

4. IHM d'énumération des versions

- affichage des n-uplets (les lignes de la table en base) de l'item concerné
- actions visualiser, restaurer (inutile de pouvoir supprimer un snapshot)
- en bonus ajouter la possibilité d'activer/désactiver une étoile pour marquer manuellement certains clichés.
- en bonus, du code javascript pour grouper (folder) par défaut les modifs faites à 5 minutes d'intervalle.

5. IHM de visualisation des snapshots : réutiliser la page des items

- (récupération de l'item depuis la table puis import XML)
- prévoir un lien vers les snapshots sur l'IHM des items
- éventuellement prévoir la suppression de l'ensemble des snapshots liés à l'item sur l'IHM des items, lors de la suppression de l'item ?
- retirer les actions de modification (ajout de blocs 'if')
- ajouter l'action spécifique : restaurer  
(le retour à l'objet peut se faire via le fil d'Ariane, donc inutile d'avoir une action de suppression de version)

**#4 - 23 décembre 2019 16:28 - Frédéric Péters**

- *Priorité changé de Bas à Normal*

**#5 - 19 mars 2020 16:27 - Mikaël Ates (de retour le 29 avril)**

- *Lié à Development #40832: Produire un journal de l'activité des utilisateurs dans le backoffice. ajouté*

**#6 - 27 mai 2020 16:54 - Frédéric Péters**

- *Echéance mis à 07 septembre 2020*

- *Assigné à mis à Frédéric Péters*

**#7 - 18 août 2020 18:48 - Frédéric Péters**

- *Statut changé de Nouveau à Rejeté*

Obsolète, traité dans [#4960](#) parce que je préférais le numéro de ticket bas.