

w.c.s. - Development #62800

action "édition", historisation des valeurs

15 mars 2022 14:47 - Frédéric Péters

| | | | |
|---|------------------|----------------------|------------------------|
| Statut: | Fermé | Début: | 15 mars 2022 |
| Priorité: | Normal | Echéance: | |
| Assigné à: | Lauréline Guérin | % réalisé: | 0% |
| Catégorie: | | Temps estimé: | 0:00 heure |
| Version cible: | | Planning: | Non |
| Patch proposed: | Oui | | |
| Description | | | |
| Du travail de spécification sur la tarification, | | | |
| Ça peut amener la nécessité d'une historisation des données côté w.c.s., parce qu'on voudra pouvoir obtenir les critères au moment de la réservation et/ou l'événement. | | | |
| Techniquement, on pourrait activer l'historisation sur les fiches, et après édition/donnée de traitement, on enregistrerait un snapshot. | | | |
| (ce ticket limité à l'enregistrement de l'info, pour l'affichage ou l'exploitation dans l'API, ça sera ailleurs). | | | |
| Demandes liées: | | | |
| Lié à w.c.s. - Development #63811: Historisation des fiches | | Fermé | 11 avril 2022 |
| Lié à w.c.s. - Development #70271: API formdata: récupérer les données à un i... | | Fermé | 14 octobre 2022 |
| Lié à w.c.s. - Development #70868: Historisation des données, afficher les mo... | | Fermé | 31 octobre 2022 |
| Lié à Publik - Development #70054: Mieux cadrer et tracer les modifications | | Fermé | 10 octobre 2022 |

Révisions associées

Révision 5fc369a0 - 31 octobre 2022 11:43 - Lauréline Guérin

formdata: store data history (#62800)

Révision b6b72033 - 31 octobre 2022 11:43 - Lauréline Guérin

formdata: look at files in history when deleting unused files (#62800)

Historique

#1 - 17 mars 2022 08:58 - Olivier Renard

Pour appui de la demande, cela permettrait de tracer l'historique de la demande

#3 - 11 avril 2022 16:28 - Frédéric Péters

- Lié à Development #63811: Historisation des fiches ajouté

#4 - 13 septembre 2022 17:43 - Frédéric Péters

Techniquement j'imaginerais au moment des actions qui peuvent modifier les données (édition, modification fiche, donnée de traitement), enregistrer une copie des données dans un objet type ContentSnapshotPart ajouté dans evolution (le plus simple étant de juste taper dedans une copie de formdata.data et laisser ça se pickler, je ne pense pas utile de faire intervenir ici de l'export/import JSON).

Petit moment d'attention à avoir par rapport à la suppression des fichiers inutilisés, et par rapport à l'anonymisation.

Ensuite pour ce que je notais pour d'autres tickets : la partie API, quand une date serait fournie il s'agirait juste de remonter dans l'historique de l'évolution voir s'il y a un ContentSnapshotPart à prendre plutôt que le formdata.data.

Et pour une partie affichage, il y aurait à pareil retrouver les données avant/après et faire un affichage basique des champs modifiés, façon redmine "Rustine proposée changé de Non à Oui".

#5 - 24 septembre 2022 18:45 - Lauréline Guérin

- Assigné à mis à Lauréline Guérin

#6 - 30 septembre 2022 13:10 - Lauréline Guérin

des notes:

- on part sur une part ContentSnapshotPart, qu'on ajoute à la dernière évolution à chaque création de formdata/carddata (par une action de workflow, lors d'une saisie, d'un import json/csv, un appel api, ...), et à chaque modification (par une action de workflow, par un appel api, par une édition dans le BO, ...)
- au départ je pensais stocker des json old_data et new_data dans la part, pour avoir le avant/après à chaque enregistrement
- nouvelle piste: avoir une table dédiée pour ces données (un peu comme la table snapshot pour les formdef et carddef), y stocker formdef_type, formdef_id, formdata_id, old_data et new_data, et dans la part pointer sur une entrée de la table. (si on perd un formdef et sa table pour une raison X ou Y, on pourra ainsi retrouver la dernière version de chaque donnée)
- stocker aussi la date de chaque changement (timestamp auto now), et un type d'event (created, updated, deleted) ?

#7 - 06 octobre 2022 15:42 - Lauréline Guérin

nouvelle piste: avoir une table dédiée pour ces données ... (si on perd un formdef et sa table pour une raison X ou Y, on pourra ainsi retrouver la dernière version de chaque donnée)

Hésitations:

- si on ne stocke qu'une copie de formdata.data, ça n'est pas suffisant pour restaurer les données en cas de perte (il manque le user, les évolutions, ...)
- et puis je me heurte à des soucis de stockage pour les objets file, et je n'ai pas encore creusé suffisamment

Pour l'anonymisation:

```
def anonymise(self):
    for field in self.formdef.get_all_fields():
        if field.anonymise:
            field.set_value(self.data, None)

    self.anonymised = datetime.datetime.now()
    self.user_id = None
    self.user_label = None
    self.editable_by = None
    self.workflow_data = None
    self.workflow_roles = None
    self.submission_context = None

    if self.evolution:
        for evo in self.evolution:
            evo.who = None
            evo.parts = None
            evo.comment = None
            evo.parts = None
    self.store()
```

Toutes les parts sont supprimées (2 fois même, on sait jamais), dans l'hypothèse où on ne fait pas de table dédiée, je laisse comme ça, ou je garde les SnapshotContentPart en anonymisant ce qui doit l'être ?

#8 - 07 octobre 2022 11:41 - Lauréline Guérin

- Fichier 0003-formdata-look-at-files-in-history-when-deleting-unus.patch ajouté
- Fichier 0002-formdata-store-data-history-62800.patch ajouté
- Fichier 0001-misc-split-workflow-tests.patch ajouté
- Statut changé de Nouveau à En cours

pour discussion

(correspond à la branche wip/62800-formdata-snapshot-bis - version sans table dédiée)

- 0001: juste un split de tests
- 0002: les ContentSnapshotPart à la création et à la modification d'un form/carddata
- 0003: fichiers inutilisés & historisation (pas très sûre de moi là)

Rien de particulier pour l'anonymisation pour le moment

#9 - 14 octobre 2022 11:43 - Lauréline Guérin

- Lié à Development #70271: API formdata: récupérer les données à un instant t ajouté

#10 - 14 octobre 2022 11:53 - Lauréline Guérin

- Fichier 0003-formdata-look-at-files-in-history-when-deleting-unus.patch ajouté
- Fichier 0002-formdata-store-data-history-62800.patch ajouté
- Fichier 0001-misc-split-workflow-tests.patch ajouté
- Statut changé de En cours à Solution proposée
- Patch proposed changé de Non à Oui

nouvelle version de 0002, pour stocker aussi formdef_type et formdef_id dans ContentSnapshotPart, parce que ça peut servir.

#13 - 19 octobre 2022 18:23 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

Toutes les parts sont supprimées (2 fois même, on sait jamais), dans l'hypothèse où on ne fait pas de table dédiée, je laisse comme ça, ou je garde les SnapshotContentPart en anonymisant ce qui doit l'être ?

Je dirai qu'arriver à l'anonymisation l'historique des données du formulaire n'ont plus beaucoup d'intérêt, donc non on vire tout.

- 0002:

```
class ContentSnapshotPart(EvolutionPart):
    def __init__(self, formdata, old_data):
        self.formdef_type = formdata.formdef.xml_root_node
        self.formdef_id = formdata.formdef.id
        self.old_data = old_data
        self.new_data = copy.deepcopy(formdata.data)
```

Est-ce qu'on ne réduirait pas un poil ? Genre ne garder que ce qui a changé :

```
- self.old_data = old_data
- self.new_data = copy.deepcopy(formdata.data)
+ self.new_data = copy.deepcopy({key: value for key, value in formdata.data.items() if value != old_data.get(key)})
```

- 0003: il me semble qu'il y manque le cas des champs fichiers dans les blocs qui est géré un peu plus haut pour formdata.data en commençant avec cette ligne :

```
elif isinstance(field_data, dict) and isinstance(field_data.get('data'), list):
```

#14 - 20 octobre 2022 11:05 - Lauréline Guérin

- Fichier 0003-formdata-look-at-files-in-history-when-deleting-unus.patch ajouté
- Fichier 0002-formdata-store-data-history-62800.patch ajouté
- Fichier 0001-misc-split-workflow-tests.patch ajouté
- Statut changé de En cours à Solution proposée

0003: il me semble qu'il y manque le cas des champs fichiers dans les blocs qui est géré un peu plus haut pour formdata.data en commençant avec cette ligne :

exact, merci

Est-ce qu'on ne réduirait pas un poil ?

C'est quand même super pratique d'avoir un avant/après complet sur une part:

- pour afficher les changements dans l'historique d'une demande, il n'y a qu'à définir une méthode view sur ContentSnapshotPart, on a tout ce qu'il faut pour faire un diff. Sinon, pour chaque part, il faudrait loader les part précédentes, construire un data complet en compilant tout ça, puis vérifier les changements.

- pour récupérer les données à un instant t, il suffit de récupérer la part juste avant t, on a les données complètes dessus. Sinon, pareil, faut récupérer toutes les parts précédentes, etc.

#15 - 20 octobre 2022 11:07 - Benjamin Dauvergne

Lauréline Guérin a écrit :

C'est quand même super pratique d'avoir un avant/après complet sur une part:

- pour afficher les changements dans l'historique d'une demande, il n'y a qu'à définir une méthode view sur ContentSnapshotPart, on a tout ce qu'il faut pour faire un diff. Sinon, pour chaque part, il faudrait loader les part précédentes, construire un data complet en compilant tout ça, puis vérifier les changements.
- pour récupérer les données à un instant t, il suffit de récupérer la part juste avant t, on a les données complètes dessus. Sinon, pareil, faut récupérer toutes les parts précédentes, etc.

Ok.

Je rejette un coup d'oeil quand c'est vert.

#16 - 20 octobre 2022 12:35 - Benjamin Dauvergne

- Statut changé de Solution proposée à Information nécessaire

Dans la modification à l'action edit_carddata pourquoi est-il nécessaire de créer une nouvelle evolution ce qui n'est pas le cas pour backoffice_field, l'édition en front ou la mise à jour par l'API ?

#17 - 20 octobre 2022 14:13 - Lauréline Guérin

- Statut changé de Information nécessaire à Solution proposée

Dans la modification à l'action edit_carddata pourquoi est-il nécessaire de créer une nouvelle evolution ce qui n'est pas le cas pour backoffice_field, l'édition en front ou la mise à jour par l'API ?

Parce que edit_carddata est lancé par une action sur un autre workflow que celui de la fiche modifiée. Cette action ajoute bien une evolution et une part sur la demande initiale, mais rien sur la fiche modifiée. Pour avoir le bon horodatage il faut ajouter une evolution, et pour comprendre d'où viennent les modifs il y a [#70000](#).

backoffice_field: on est dans un perform_workflow, qui gère déjà les evolutions et les ActionsTracingEvolutionPart et l'horodatage des actions.
édition en front: idem, on passe par un perform_workflow
api: juste avant l'ajout de la part, on a un perform_workflow.
Il me semble qu'on est bon.

En tout cas les tests ajoutés sur evolution et parts dans test_content_snapshots.py m'ont l'air plutôt cohérents, mais je veux bien une autre relecture pour confirmer.

#19 - 20 octobre 2022 15:56 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Ok.

#20 - 21 octobre 2022 12:12 - Lauréline Guérin

- Fichier 0003-formdata-look-at-files-in-history-when-deleting-unus.patch ajouté
- Fichier 0002-formdata-store-data-history-62800.patch ajouté
- Fichier 0001-misc-split-workflow-tests.patch ajouté
- Statut changé de Solution validée à Solution proposée

Après discussion avec Fred: ajout d'une datetime sur ContentSnapshotPart, pour avoir la bonne date (avec tz) même si la part est posée sur la mauvaise evolution

#21 - 24 octobre 2022 17:36 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Toujours ok.

#22 - 31 octobre 2022 11:41 - Lauréline Guérin

- Lié à Development #70868: Historisation des données, afficher les modifications dans l'historique d'une demande/fiche ajouté

#23 - 31 octobre 2022 12:11 - Lauréline Guérin

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

```
commit b6b7203358a355a950bde26deb96dd9166379b68
Author: Lauréline Guérin <zebuline@entrouvert.com>
Date: Fri Oct 7 09:40:16 2022 +0200
```

formdata: look at files in history when deleting unused files (#62800)

```
commit 5fc369a082124f334f5fcb609fe35417c2b31b08
Author: Lauréline Guérin <zebuline@entrouvert.com>
Date: Sat Sep 24 18:43:55 2022 +0200
```

formdata: store data history (#62800)

```
commit f293184fff301048fe9d21c5d552f5f1960ae080
Author: Lauréline Guérin <zebuline@entrouvert.com>
Date: Sat Sep 24 18:35:59 2022 +0200
```

misc: split workflow tests

#24 - 31 octobre 2022 20:14 - Transition automatique

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

#25 - 01 janvier 2023 04:41 - Transition automatique

Automatic expiration

#27 - 13 avril 2023 16:31 - Mikaël Ates (de retour le 29 avril)

- Lié à Development #70054: Mieux cadrer et tracer les modifications ajouté

Fichiers

| | | | |
|---|---------|-----------------|------------------|
| 0003-formdata-look-at-files-in-history-when-deleting-unus.patch | 8,52 ko | 07 octobre 2022 | Lauréline Guérin |
| 0002-formdata-store-data-history-62800.patch | 45,6 ko | 07 octobre 2022 | Lauréline Guérin |
| 0001-misc-split-workflow-tests.patch | 103 ko | 07 octobre 2022 | Lauréline Guérin |
| 0003-formdata-look-at-files-in-history-when-deleting-unus.patch | 8,52 ko | 14 octobre 2022 | Lauréline Guérin |
| 0002-formdata-store-data-history-62800.patch | 49 ko | 14 octobre 2022 | Lauréline Guérin |
| 0001-misc-split-workflow-tests.patch | 103 ko | 14 octobre 2022 | Lauréline Guérin |
| 0003-formdata-look-at-files-in-history-when-deleting-unus.patch | 11,8 ko | 20 octobre 2022 | Lauréline Guérin |
| 0001-misc-split-workflow-tests.patch | 103 ko | 20 octobre 2022 | Lauréline Guérin |
| 0002-formdata-store-data-history-62800.patch | 49 ko | 20 octobre 2022 | Lauréline Guérin |
| 0003-formdata-look-at-files-in-history-when-deleting-unus.patch | 11,8 ko | 21 octobre 2022 | Lauréline Guérin |
| 0001-misc-split-workflow-tests.patch | 103 ko | 21 octobre 2022 | Lauréline Guérin |
| 0002-formdata-store-data-history-62800.patch | 50,2 ko | 21 octobre 2022 | Lauréline Guérin |