

w.c.s. - Development #53253

API, inclure dans le json d'une demande les fichiers qui sont dans l'historique

19 avril 2021 14:51 - Frédéric Péters

Statut:	Fermé	Début:	19 avril 2021
Priorité:	Normal	Echéance:	05 mai 2021
Assigné à:	Nicolas Roche	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Sur l'export JSON d'une demande on passe sur l'historique et dedans,

```
for part in self.parts or []:  
    if hasattr(part, 'get_json_export_dict'):  
        parts.append(part.get_json_export_dict (anonymise=anonymise))
```

mais AttachmentEvolutionPart n'a pas de méthode get_json_export_dict et résultat un fichier attaché ainsi ne se trouve pas repris.

Il y aurait à passer la valeur de include_files jusque là et qu'un et_json_export_dict soit ajouté à l'objet.

Demandes liées:

Dupliqué par w.c.s. - Development #39015: api: exposer les fichiers joints à ... **Fermé** **16 janvier 2020**

Révisions associées

Révision 51d93565 - 05 mai 2021 22:30 - Nicolas Roche

api: include attachment evolution parts in {card,form}data fetch API (#53253)

Historique

#2 - 20 avril 2021 17:08 - Frédéric Péters

- Echéance mis à 05 mai 2021

#4 - 26 avril 2021 14:33 - Nicolas Roche

- Assigné à mis à Nicolas Roche

#5 - 27 avril 2021 17:07 - Nicolas Roche

- Fichier 0001-api-add-attachment-evolution-parts-to-json-53253.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Je propose ce patch bien que j'ai l'impression d'avoir raté quelque chose : je n'ai pas trouvé de fonction adéquat sur laquelle m'appuyer pour éviter d'ajouter du code.

Voici un exemple de rendu (j'ai inventé un type "workflow-attachment") :

```
{...  
  "evolution": [  
    {...  
      "parts": [  
        {  
          "content": "/9j/4AAQSkZJ...",  
          "content_type": "image/jpeg",  
          "filename": "01_Crop_Circles_Intricate.jpg",  
          "type": "workflow-attachment"  
        }  
      ]  
    }  
  ],  
}
```

Le test aussi me semble un peu léger : il ne fait pas explicitement l'appel aux actions "Fichier joint" et "Message dans l'historique", parce que je me suis inspiré des autres tests trouvés dans tests/api/ et que je ne vois pas où en ajouter ailleurs.

Aussi, j'ai propagé le paramètre 'include_files' (j'ai cru comprendre que c'était ce qui est demandé dans le ticket) mais je n'y trouve pas grand intérêt : je n'ai pas vu comment le positionner à False via l'api.

#6 - 27 avril 2021 19:01 - Frédéric Péters

include_files doit être à True par défaut partout; la signature de la méthode get_json_export_dict doit être la même partout.

On a le déroulé include_files=False quand l'API /api/forms/<slug>/, ou /api/users/forms sont appelées, avec ?full=on, exemple:

```
def get_formdata_dict(formdata, user, consider_status_visibility=True):
    ...
    if get_request().form.get('full') == 'on':
        d.update(formdata.get_json_export_dict(include_files=False, user=user))
```

Voilà donc un test supplémentaire à ajouter : assurer que tu ne reçois pas l'information sur les fichiers dans ces appels.

Je n'ai pas cherché où tu trouvais ton inspiration pour l'anonymisation qui conserve le nom de fichier; je veux bien croire en cette possibilité.

```
if not anonymise:
    d['content'] = base64.encodebytes(open(self.filename, 'rb').read())
```

utiliser with pour assurer le nettoyage du descripteur de fichier.

#7 - 27 avril 2021 20:28 - Nicolas Roche

Je n'ai pas cherché où tu trouvais ton inspiration pour l'anonymisation qui conserve le nom de fichier

non, ça c'est sorti tout droit de ma tête, je vais le retirer.

(merci pour '?full=on', j'avais mal cherché)

#8 - 28 avril 2021 12:45 - Nicolas Roche

- Fichier 0004-api-add-attachment-evolution-parts-to-json-53253.patch ajouté
- Fichier 0003-misc-normalize-export_to_json-parameters-53253.patch ajouté
- Fichier 0002-misc-normalize-get_json_export_dict-parameters-53253.patch ajouté
- Fichier 0001-misc-reorder-get_json_export_dict-parameters-53253.patch ajouté

(remarques prisent en compte)

la signature de la méthode get_json_export_dict doit être la même partout.

oui, j'avais hésité là dessus (j'ai pris peur d'ouvrir la boite de pandore)

#9 - 28 avril 2021 13:48 - Frédéric Péters

oui, j'avais hésité là dessus (j'ai pris peur d'ouvrir la boite de pandore)

Houla clairement quand j'écris ça c'est pour que tu ne mettes pas un =False alors qu'ailleurs c'est =True, ou l'inverse; surtout ne vient pas modifier de l'existant. (je viens de regarder 0001).

#10 - 28 avril 2021 14:05 - Frédéric Péters

Bref, le patch initial, sans la confusion un coup =True un coup =False, et le test que je mentionnais. Et ne rien reprendre en anonyme.

#11 - 28 avril 2021 15:43 - Nicolas Roche

- Fichier 0001-api-add-attachment-evolution-parts-to-json-53253.patch ajouté

Remarques prises en compte,

utiliser with pour assurer le nettoyage du descripteur de fichier.

y compris celle là que j'avais oublié de traiter.

#12 - 04 mai 2021 14:17 - Frédéric Péters

J'ai encore du manquer un bout de mon explication sur les signatures.

```
- def get_json_export_dict(self, user, anonymise=False):
+ def get_json_export_dict(self, user, include_files=True, anonymise=False):
```

Il faut ajouter les nouveaux paramètres derrière les paramètres existants.

```
if get_request().form.get('full') == 'on':
    include_files = False
else:
    include_files = True
```

tu pourrais juste passer `bool(get_request().form.get('full') == 'on')`.

À part ça je pense que ça va être ok.

#13 - 04 mai 2021 15:16 - Nicolas Roche

- *Statut changé de Solution proposée à En cours*

Il faut ajouter les nouveaux paramètres derrière les paramètres existants.

Même si la nouvelle signature existe déjà par ailleurs ?

```
$ git grep -n 'def get_json_export_dict'
wcs/formdata.py:1145: def get_json_export_dict(self, include_files=True, anonymise=False, user=None):
```

#14 - 04 mai 2021 15:21 - Frédéric Péters

Oui, fais un patch qui se tienne pour lui.

#15 - 04 mai 2021 17:21 - Nicolas Roche

- *Fichier 0001-api-add-attachment-evolution-parts-to-json-53253.patch ajouté*

- *Statut changé de En cours à Solution proposée*

(Remarques prises en compte.)

#16 - 05 mai 2021 22:27 - Frédéric Péters

- *Fichier 0001-api-include-attachment-evolution-parts-in-card-form-.patch ajouté*

tu pourrais juste passer `bool(get_request().form.get('full') == 'on')`.

J'écrivais ça et le résultat c'est

```
+ anonymise=anonymise, include_files=not bool(get_request().form.get('full') == 'on')
```

et ça me sonne moche comme tout (le `not` qui vient se caler là, plutôt que modifier la comparaison) et je m'arrête je m'arrête et j'ai totalement besoin d'une explication de comment on est arrivé à dire que passer `full=on` ça voudrait dire ne pas avoir les fichiers, ça me semble totalement à l'envers ?

..

Et je relis diligemment le tout et il y a eu confusion entre le comportement de l'API qui retourne l'ensemble des fiches (là le `full=on` il est pour dire "tous les attributs") et l'API qui retourne les données d'une fiche, là il n'y a pas d'option au comportement.

Il y a un test qui fait pourtant ça. Je viens de le corriger.

```
--- a/tests/api/test_formdata.py
+++ b/tests/api/test_formdata.py
@@ -713,7 +713,7 @@ def test_api_anonymized_formdata(pub, local_user, admin_user):
     assert 'who' not in resp.json[0]['evolution'][0]
     assert 'time' in resp.json[0]['evolution'][0]
     # check anonymise is enforced on detail view
- resp = get_app(pub).get(sign_uri('/api/forms/test/%s/?anonymise&full=on' % resp.json[1]['id']))
```

```
+ resp = get_app(pub).get(sign_uri('/api/forms/test/%s/?anonymise' % resp.json[1]['id']))
+ assert 'receipt_time' in resp.json
+ assert 'fields' in resp.json
+ assert 'user' not in resp.json
```

~~

Cela posé, ça permet de retirer du patch la partie qui touche à `wcs/forms/common.py`, et donc la partie dont heureusement le style m'a heurté 2 fois parce que même la première fois je n'avais pas noté le non-sens.

~~

Et à reprendre :

On a le déroulé `include_files=False` quand l'API `/api/forms/<slug>/`, ou `/api/users/forms` sont appelées, avec `?full=on`, exemple:

...

Voilà donc un test supplémentaire à ajouter : assurer que tu ne reçois pas l'information sur les fichiers dans ces appels.

Et donc comme il y a eu confusion sur l'API (j'écris `/api/forms/<slug>/` et c'est `/api/forms/<slug>/<id>` qui est traité), j'ajoute le test dont je parlais,

```
@@ -514,6 +514,10 @@ def test_formdata_with_evolution_part_attachment(pub, local_user):
+     assert len(resp.json['evolution']) == 1
+     assert 'parts' not in resp.json['evolution']
```

```
+ # check this doesn't get into list of forms API
+ resp = get_app(pub).get(sign_uri('/api/forms/test/list?full=on', user=local_user))
+ assert 'hello.txt' not in resp.text
```

(oui erreur de ma part l'API c'était `/api/forms/<slug>/list` et pas `/api/forms/<slug>/`).

~~

Et tout ça amène à quelque chose qui était presque le premier patch, qui avait juste comme erreur de mettre `include_files=False` plutôt que `True`.

Et mon commentaire qui suivait :

`include_files` doit être à `True` par défaut partout; la signature de la méthode `get_json_export_dict` doit être la même partout.

Avec la partie sur la signature qui voulait pointer que dans le patch tu avais à la fois :

```
def get_json_export_dict(self, include_files=True, anonymise=False):
```

et

```
def get_json_export_dict(self, include_files=False, anonymise=False):
```

Puis mauvaise interprétation de mon "la signature doit partout être pareille", appliquée au-delà du patch.

Puis je reprends :

Bref, le patch initial, sans la confusion un coup `=True` un coup `=False`, et le test que je mentionnais. Et ne rien reprendre en anonyme.

Et c'est là sans doute que "le test que je mentionnais" (sur le `?full=on`) se trouve amener l'idée qu'il y a un `?full=on` sur l'API d'une demande unique, et donc l'ajout du non-sens (`?full=on` → moins de données).

~~

Voilà beaucoup écrit mais je trouve crucial d'identifier ces moments de mauvaise explication&compréhension, pour tenter de les réduire.

(patch dans la branche -bis)

#17 - 05 mai 2021 22:31 - Frédéric Péters

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

(et cette relecture fait office de validation du patch modifié, parce qu'à la base c'est pour l'avoir dans un tag que je me suis remis à relire ça ce soir)

```
commit 51d935656a8105963e7afff428d402cfb37c7758
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Tue Apr 27 11:15:01 2021 +0200
```

```
api: include attachment evolution parts in {card,form}data fetch API (#53253)
```

#18 - 05 mai 2021 23:16 - Frédéric Péters

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

#19 - 15 novembre 2023 20:03 - Frédéric Péters

- Dupliqué par Development #39015: api: exposer les fichiers joints à une demande au cours du workflow ajouté

Fichiers

0001-api-add-attachment-evolution-parts-to-json-53253.patch	9,4 ko	27 avril 2021	Nicolas Roche
0004-api-add-attachment-evolution-parts-to-json-53253.patch	9,17 ko	28 avril 2021	Nicolas Roche
0003-misc-normalize-export_to_json-parameters-53253.patch	6,76 ko	28 avril 2021	Nicolas Roche
0002-misc-normalize-get_json_export_dict-parameters-53253.patch	6,82 ko	28 avril 2021	Nicolas Roche
0001-misc-reorder-get_json_export_dict-parameters-53253.patch	1,92 ko	28 avril 2021	Nicolas Roche
0001-api-add-attachment-evolution-parts-to-json-53253.patch	11,5 ko	28 avril 2021	Nicolas Roche
0001-api-add-attachment-evolution-parts-to-json-53253.patch	11,5 ko	04 mai 2021	Nicolas Roche
0001-api-include-attachment-evolution-parts-in-card-form-.patch	6,53 ko	05 mai 2021	Frédéric Péters