Passerelle - Development #29713

csv : endpoint de mise à jour du fichier

14 janvier 2019 14:05 - Emmanuel Cazenave

Statut:	Fermé	**Début:**	14 janvier 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Nicolas Roche	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Un cas d'usage : #29705

Révisions associées

Révision 593a7850 - 09 juillet 2021 10:27 - Nicolas Roche

views: implement GenericEndpointView.put (#29713)

Révision ec7b5c51 - 09 juillet 2021 10:27 - Nicolas Roche

csvdatasource: add upload-csv-file endpoint (#29713)

Historique

#1 - 21 juin 2021 19:00 - Frédéric Péters

- Assigné à mis à Nicolas Roche

- Planning mis à Non

#2 - 22 juin 2021 22:31 - Nicolas Roche

- Fichier 0001-csvdatasource-add-upload-csv-file-endpoint-29713.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

ex:

```
$ curl https://passerelle.dev.publik.love/csvdatasource/test/update -d'{"filename": "data.csv", "content": "tr
uc encodé en base 64"}'
```

J'ai préféré faire en post (comme c'est fait ailleurs) plutôt que comme c'est indiqué das le tickét lié, via

```
$ curl https://passerelle.dev.publik.love/csvdatasource/test/update -T ~/Téléchargements/data.csv
```

parce que :

- ' curl -T ' ne passe pas de nom du fichier (pas de multipart juste du contenu brut)
- je n'arrive pas à écrire (via django-webtest) un test qui simule ' curl -T '

J'ai forcé le passage du nom de fichier pour ne pas avoir tenter de deviner le nom du précédent fichier (en retirant les _xxxxxxx).
J'écrase le nom de la feuille de style avec la chaîne vide si elle n'est pas fournie, pour ne pas introduire de magie.
J'ai placé ce nouveau endpoint en premier (display_order=-1) pour faciliter l'usage du gabarit endpoint.html.
J'ai copié/collé le schéma JSON du connecteur CMIS (avec ses regexes).

#3 - 22 juin 2021 22:35 - Nicolas Roche

- Fichier 0001-csvdatasource-add-upload-csv-file-endpoint-29713.patch ajouté

(j'ai retiré un import que j'ai introduit à tord dans le précédent patch)

#4 - 23 juin 2021 01:42 - Thomas Noël

- Statut changé de Solution proposée à En cours

C'est bien trop compliqué à utiliser, il faut faire un endpoint qui mange un fichier envoyé classiquement et sera utilisable directement avec cURL :

```
curl -X POST/PUT/PATCH -H "Content-Type: text/csv ou application/vnd.oasis.opendocument.spreadsheet ou autre"
https://passerelle/csvdatasource/slug/update-file?apikey=xxxx --data-binary @/tmp/le-fichier-machin.ods
```

Pas de mise à jour du sheet_name, juste le contenu du fichier, qui doit rester exactement au même format.

Au passage, prévoir une permission spéciale perm='can_update_file' plutôt que le simple can_access.

#5 - 23 juin 2021 01:51 - Thomas Noël

A noter que :

- ' curl -T ' ne passe pas de nom du fichier (pas de multipart juste du contenu brute)

C'est pas grave, tu reprends le nom du fichier existant ou tu laisses Django se débrouiller avec la mise à jour de csv_file, et si vraiment t'arrives à rien tu mets "api-uploaded-file-<aaaammddhhmmss>.<ext>"

#6 - 23 juin 2021 09:06 - Benjamin Dauvergne

Une remarque en passant : un intérêt à supporter un format {"file": {"base64_content"...}} en plus de l'upload direct (et on peut supporter les deux en se basant sur la condition Content-Type: application/json) c'est que ça permet de gérer ça depuis un formulaire w.c.s. "Mise à jour du référentiel XXXX", qui nous donne un historique des actions.

#7 - 23 juin 2021 09:28 - Nicolas Roche

Ok, ce sera l'occasion d'implémenter le PUT dans passerelle,
et aussi l'authentification basique non, si on veut pousser la simplification au bout ?
(gros ticket en fait)

#8 - 23 juin 2021 10:27 - Benjamin Dauvergne

Nicolas Roche a écrit :

> Ok, ce sera l'occasion d'implémenter le PUT dans passerelle,

Il y a déjà un PUT dans le connecteur Greco; mais il y a surtout des soucis pour la doc des schémas et les descriptions quand on déclare plusieurs méthodes sur un même endpoint, voir le commentaire "BUG" dans passerelle/contrib/rsa13/models.py. La seule conséquence c'est qu'il n'est pas évident de faire une API Rest dans le style une ressource/plusieurs méthodes comme on fait avec DRF. Mais là le besoin ne me semble pas être ça.

> et aussi l'authentification basique non, si on veut pousser la simplification au bout ?
> (gros ticket en fait)

Non apikey est rigoureusement équivalent à HTTP Basic en terme de sécu (c'est un secret partagé en clair) et ça demande aucun boulot, juste mettre ?apikey= dans l'URL (l'inconvénient c'est que l'apikey est visible dans les logs des proxys, je pense qu'on peut continuer à vivre avec, ils voient tout dans l'absolu, mais ne le log que si demandé pour le reste); donc aucun besoin d'HTTP Basic, ça n'a jamais bloqué personne à ma connaissance.

#9 - 23 juin 2021 10:52 - Nicolas Roche

> Il y a déjà un PUT dans le connecteur Greco

J'ai un doute sur le fait qu'il fonctionne : il manque le put dans passerelle/views.py::GenericEndpointView
https://dev.entrouvert.org/issues/51983 (il manque juste les 2 lignes, pas tout le patch)

> Non apikey est rigoureusement équivalent à HTTP Basic en terme de sécu (c'est un secret partagé en clair)

Merci, je n'avais pas encore compris comment ça fonctionnait.

Donc ici :

- retirer l'envoie du nom du fichier (en inventer un) et du nom de la feuille de style
- supporter les 2 formats 'curl -T' et le json avec base 64
- prévoir une permission spéciale perm='can_update_file'

Mais je n'ai pas compris si "qui doit" est quelque-chose que je dois assurer ou juste supposer :

- juste (la mise à jour) le contenu du fichier, qui doit rester exactement au même format.

#10 - 23 juin 2021 10:57 - Benjamin Dauvergne

Nicolas Roche a écrit :

> https://dev.entrouvert.org/issues/51983 (il manque juste les 2 lignes, pas tout le patch)

Tu dois avoir raison.

> Donc ici :
>
> - retirer l'envoie du nom du fichier (en inventer un) et du nom de la feuille de style

ou conserver le nom du fichier actuel s'il y en a déjà un.

> - supporter les 2 formats 'curl -T' et le json avec base 64

Oui je pense que ça ne mange pas de pain de faire ça en se basant sur le content-type quand il vaut "application/json".

> - prévoir une permission spéciale perm='can_update_file'
>
> Mais je n'ai pas compris si "qui doit" est quelque-chose que je dois assurer ou juste supposer :
>
> - juste (la mise à jour) le contenu du fichier, qui doit rester exactement au même format.

Je crois que Thomas veut dire ne faire aucun contrôle à part ce que l'on ferait déjà sur un upload sur le backoffice (je ne sais pas si on fait un contrôle sur les colonnes, ou au moins sur leur nombre).

#11 - 23 juin 2021 11:58 - Thomas Noël

Benjamin Dauvergne a écrit :

> Nicolas Roche a écrit :
>
> > https://dev.entrouvert.org/issues/51983 (il manque juste les 2 lignes, pas tout le patch)
>
> Tu dois avoir raison.

Ne pars pas dans des sous-sous-tickets : si PUT ne marche pas, alors gère juste POST, basta.

> Je crois que Thomas veut dire ne faire aucun contrôle à part ce que l'on ferait déjà sur un upload sur le backoffice

Indeed.

#12 - 23 juin 2021 15:08 - Nicolas Roche

> Ne pars pas dans des sous-sous-tickets : si PUT ne marche pas, alors gère juste POST, basta.

si j'ai parlé de ça c'est parce que 'curl -T' fait un PUT.

Donc je raye ?

- supporter ~~les 2 formats 'curl -T' et le~~ json avec base 64

edit:

> ou conserver le nom du fichier actuel s'il y en a déjà un.

En fait on n'écrase pas les fichiers mais on les effaces via cron.
Donc si j'ai data.csv, et que je repousse data.csv alors j'aurais en fait data_azAZ567.csv
Et donc il me faudra deviner data.csv depuis data_azAZ567.csv (ok mais c'est du code moche).

> aucun contrôle à part ce que l'on ferait déjà sur un upload sur le backoffice

Via le backoffice, on ne peut pas pousser de fichier ODS sans indiquer de nom de feuille.

Ici il faudra donc refuser le chargement d'un ODS si la feuille n'est pas renseignée dans le connecteur,
(ou peut-être si le format CSV,ODS... détecté par le connecteur diffère).

#13 - 23 juin 2021 15:31 - Frédéric Péters

> Et donc il me faudra deviner data.csv depuis data_azAZ567.csv (ok mais c'est du code moche).

Non tu inventes rien on s'en tape du nom de fichier.

Et pareil pour le PUT, soit ça marche et ok, soit ça n'existe pas encore dans Passerelle et c'est un autre ticket et le sujet disparait de celui-ci.

Dispersion zéro.

#14 - 23 juin 2021 16:10 - Nicolas Roche

Donc ici :

- retirer l'envoie du nom du fichier (en inventer un) et du nom de la feuille de style
- refuser le chargement d'un ODS si la feuille n'est pas renseignée dans le connecteur
- supporter le json json avec base 64

Pour le 2 premiers points je peux aussi négocier ?
(parce que là on retomberait sur mon patch qui le mérite de ne pas ajouter de code spécifique)

Quoi qu'il en soit il me restera :

- prévoir une permission spéciale perm='can_update_file'

#15 - 23 juin 2021 16:24 - Frédéric Péters

Tu vas dans wcs, tu crées un formulaire avec un champ fichier, tu lui donnes comme identifiant "fichier". Tu crées un workflow, tu associes le workflow au formulaire, dans le workflow tu crées une action "appel webservice". Tu configures cette action pour être un POST et envoyer un paramètre nommé file et avec comme valeur {{form_var_fichier}}.

Tu mets pour cet appel l'URL vers le nouvel endpoint.

Ensuite tu publies le formulaire tu mets un fichier csv dans le champ tu valides la demande.

Par cette action le connecteur dans Passerelle doit désormais être configuré avec ce nouveau fichier.

#16 - 23 juin 2021 17:11 - Thomas Noël

Frédéric Péters a écrit :

> Tu vas dans wcs (...)

Alors je ne sais PAS DU TOUT pourquoi on parle de w.c.s.

Il s'agit ici de permettre à un tiers (un client, un éditeur) de se faire un script à sa façon pour mettre à jour la nuit le csv_file d'un csvdatasource. Zéro usage de wcs.

Et pour moi, on devrait chercher à suivre le même protocole (la même API) que le import-csv des fiches dans wcs. Histoire de ne pas avoir mille doc à écrire, et rester sur un exemple facile avec cURL.

#17 - 23 juin 2021 17:15 - Frédéric Péters

> Alors je ne sais PAS DU TOUT pourquoi on parle de w.c.s.

Pour fournir à Nicolas un format à suivre et qu'il soit produit quelque chose qui fonctionne de bout en bout.

Pour oublier ce ticket le côté PUT, qui est le truc utile, mais visiblement nébuleux, et que cette partie arrive derrière, dans un nouveau ticket.

#19 - 23 juin 2021 17:34 - Thomas Noël

- Patch proposed changé de Oui à Non

#22 - 25 juin 2021 15:23 - Nicolas Roche

- Fichier 0002-csvdatasource-add-upload-csv-file-endpoint-29713.patch ajouté

- Fichier 0001-views-implement-GenericEndpointView.put-29713.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

```
curl -X PUT -H "Content-Type: text/csv ou application/vnd.oasis.opendocument.spreadsheet ou autre" https://pas
serelle/csvdatasource/slug/update-file?apikey=xxxx --data-binary @/tmp/le-fichier-machin.ods
```

Au passage, une permission spéciale perm='can_update_file' plutôt que le simple can_access.

#23 - 25 juin 2021 16:43 - Thomas Noël

"CSV file" c'est historique, le connecteur parle de "Spreadsheet file" car il gère aussi ods, xlsx, etc. Il faudrait donc modifier les phrases comme "Uploading CSV file is limited to the following API users:" ou "Modify CSV file" et parler de "Spreadsheet file".

'"accept curl -T queries"' : commentaire pas très clair et donc un peu inutile à mon avis, plutôt dire que le payload est le fichier. Ou ne rien dire, le code est assez clair je trouve (et il y a ce ticket en référence git blame)

Utilise plutôt request.content_type tu te fatigueras moins (https://docs.djangoproject.com/fr/2.2/ref/request-response/#django.http.HttpRequest.content_type)

Je ne comprends pas « storage = getattr(self.csv_file, 'storage') » ... pourquoi getattr ?

display_order=-1 : je préférerais que ce endpoint soit plutôt référencé en dernier, et dans un catégorie "Management" (et les autres dans une catégorie "Access")

#24 - 25 juin 2021 19:36 - Nicolas Roche

- Fichier 0001-csvdatasource-add-upload-csv-file-endpoint-29713.patch ajouté

> modifier les phrases comme ... et parler de "Spreadsheet file".

oups

> '"accept curl -T queries"' : inutile à mon avis

viré

> Utilise plutôt request.content_type tu te fatigueras moins

merci

> Je ne comprends pas « storage = getattr(self.csv_file, 'storage') » ... pourquoi getattr ?

c'est juste pour pylint
https://jenkins.entrouvert.org/job/passerelle-wip/job/wip%252F29713-csvdatasource-upload-endpoint/5/pylint/new/folder.-509596225/source.3a621aa2-e6cd-4e68-a3d1-d510c90c65bf/#315

> display_order=-1 : je préférerais que ce endpoint soit plutôt référencé en dernier, et dans un catégorie "Management" (et les autres dans une catégorie "Access")

fait ; j'ai du changer les libellés pour ne pas avoir les ':' en double.

#25 - 28 juin 2021 12:00 - Thomas Noël

Nicolas Roche a écrit :

> > Je ne comprends pas « storage = getattr(self.csv_file, 'storage') » ... pourquoi getattr ?
>
> c'est juste pour pylint

Vu... laisse pleurer pylint, c'est lui qui se trompe.

display_order=-1 : je préférerais que ce endpoint soit plutôt référencé en dernier, et dans un catégorie "Management" (et les autres dans une catégorie "Access")

fait ; j'ai du changer les libellés pour ne pas avoir les ':' en double.

Ok vu... bon, modifie juste le csvdatasource_detail.html pour ajouter ton endpoint en dessous des autres, et hop... un autre ticket un jour "nettoiera" ce connecteur pour le conformer à des endpoints mieux autodocumentés.

#26 - 28 juin 2021 14:26 - Nicolas Roche

- Fichier Screenshot_2021-06-28 Passerelle.png ajouté

Remarques prises en compte.
ex:

```
$ curl -H 'Content-Type: text/csv' https://passerelle.dev.publik.love/csvdatasource/test/update -T ~/Télécharg
ements/data.csv
{"created": "csvdatasource/test/api-uploaded-file_goKLCS2.csv", "err": 0}
```

#27 - 28 juin 2021 14:27 - Nicolas Roche

- Fichier 0001-views-implement-GenericEndpointView.put-29713.patch ajouté

- Fichier 0002-csvdatasource-add-upload-csv-file-endpoint-29713.patch ajouté

#28 - 28 juin 2021 15:02 - Thomas Noël

Il manque l'i18n dans le html (ie ajouter des {% trans ...)

#29 - 28 juin 2021 16:48 - Nicolas Roche

- Fichier 0002-csvdatasource-add-upload-csv-file-endpoint-29713.patch ajouté

Oui, pardon.

#30 - 05 juillet 2021 10:20 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

#32 - 09 juillet 2021 10:30 - Nicolas Roche

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

```
commit ec7b5c51cde692c2656229bc663d531e46c96076
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date:   Tue Jun 22 16:17:55 2021 +0200

    csvdatasource: add upload-csv-file endpoint (#29713)

commit 593a78504a5b855c19cadd9b47e59b86878e5400
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date:   Fri Jun 25 14:17:30 2021 +0200

    views: implement GenericEndpointView.put (#29713)
```

#33 - 13 juillet 2021 16:17 - Frédéric Péters

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

Fichiers

0001-csvdatasource-add-upload-csv-file-endpoint-29713.patch	9,79 ko	22 juin 2021	Nicolas Roche
0001-csvdatasource-add-upload-csv-file-endpoint-29713.patch	9,75 ko	22 juin 2021	Nicolas Roche
0001-views-implement-GenericEndpointView.put-29713.patch	1,3 ko	25 juin 2021	Nicolas Roche
0002-csvdatasource-add-upload-csv-file-endpoint-29713.patch	10,7 ko	25 juin 2021	Nicolas Roche
0001-csvdatasource-add-upload-csv-file-endpoint-29713.patch	13,2 ko	25 juin 2021	Nicolas Roche
Screenshot_2021-06-28 Passerelle.png	43 ko	28 juin 2021	Nicolas Roche
0001-views-implement-GenericEndpointView.put-29713.patch	1,3 ko	28 juin 2021	Nicolas Roche

0002-csvdatasource-add-upload-csv-file-endpoint-29713.patch	11,1 ko	28 juin 2021	Nicolas Roche
0002-csvdatasource-add-upload-csv-file-endpoint-29713.patch	11,1 ko	28 juin 2021	Nicolas Roche