

Authentic 2 - Development #57567

ajouter un paramètre full à l'api de synchronisation

04 octobre 2021 16:31 - Paul Marillonnet

Statut:	Fermé	Début:	04 octobre 2021
Priorité:	Normal	Echéance:	
Assigné à:	Paul Marillonnet	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
<p>plutôt qu'un seul uuid à la fois comme c'est le cas actuellement. l'objectif est #45276 où on aurait via 57564 la liste des usagers modifiés, dont on viendrait ensuite ici récupérer les détails pour appliquer les modifications dans les autres briques.</p> <p>le paramètre full permettrait de retourner les utilisateurs complets, pas simplement les UUIDs.</p>			
Demandes liées:			
Lié à Hobo - Development #45276: commande de resynchronisation des utilisateurs		En cours	20 juillet 2020
Lié à Authentic 2 - Development #58312: inclure les nouveaux usagers au param...		Nouveau	02 novembre 2021

Révisions associées

Révision 42f2571f - 02 novembre 2021 12:01 - Paul Marillonnet

api: add a full_known_users option to /synchronization/ endpoint (#57567)

Historique

#1 - 04 octobre 2021 16:33 - Paul Marillonnet

- Lié à [Development #45276](#): commande de resynchronisation des utilisateurs ajouté

#2 - 05 octobre 2021 10:47 - Paul Marillonnet

On fait déjà ça quelque part dans Publik ?

J'imaginai ce comportement enclenché par une requête GET sur /api/users/?uuids=<uuid1>,<uuid2>,<uuid3>,... mais il y a peut-être une approche plus élégante.

#3 - 06 octobre 2021 10:19 - Benjamin Dauvergne

Réponse générale :

Pour une querystring, la manière de passer des arguments multiples c'est généralement de passer plusieurs fois le paramètre (en php c'est souvent codifié en ajoutant [] au nom du paramètre), donc ici ce serait ?uuids=<uuid1>&uuids=<uuid2>&....

Réponse particulière :

Non c'est vrai qu'on n'a pas ça ailleurs, mais on a pas deux web-services pareil à part la structure des réponses, on a jamais vraiment codifié la forme des requêtes. Dans ma réponse sur la commande je propose aussi de toute faire dans synchronization si tu veux, i.e. d'ajouter un paramètre full=true par exemple qui au lieu de simplement renvoyer les uuids renvoie les utilisateurs complets, avec un minimum de pagination.

Une autre solution serait de proposer une API à part pour faire des trucs "batch" /api/batch/, inspiré de SCIM, parce qu'avec la query-string on sera limité sur le nombre d'uuid qu'on peut passer (à 38 caractères par paramètre uuid, et disons une limite d'URL à 2Ko, on pourra en passer pas beaucoup plus de 50, ce qui niveau "batching" n'est pas terrible.

```
POST /api/batch/
{
  "atomic": False, (valeur par défaut)
  "requests": [
    {
      "path": "/api/user/",
      "query": {"uuid": "...."},
    },
    ....
  ]
}
```

c'est un peu pédestre mais c'est clair au moins.

Encore une autre possibilité c'est que si `&modified_gt=..` prend bien en compte les changements sur les rôles, on a besoin de rien d'autre, `/api/synchronization/` n'a à qu'à gérer un `&since=...` pour les suppressions d'utilisateurs, `/api/user/` peut gérer tous les changements.

Ou encore on ajoute un `include-deleted=true` à `/api/user/` qui renverrait des enregistrements de la forme `{"uuid": "...", "deleted": "<date-de-suppression>"}`.

Voilà, choisis ton poison :)

#4 - 22 octobre 2021 10:19 - Paul Marillonnet

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

#5 - 22 octobre 2021 10:20 - Paul Marillonnet

- Sujet changé de *étendre /api/users/ pour qu'elle puisse requêter une liste d'uuids* à *ajouter un paramètre full à l'api de synchronisation*

Parce que c'est pour la synchro et que c'est la solution qui me semble encore la plus simple.

#6 - 22 octobre 2021 10:22 - Paul Marillonnet

- Description mis à jour

#7 - 22 octobre 2021 11:07 - Paul Marillonnet

J'ai poussé un commit de ce genre dans la branche, je regarde pour la pagination après lecture de la page de doc DRF à ce sujet avec lequel je ne suis pas très familier.

#8 - 25 octobre 2021 15:02 - Paul Marillonnet

- Fichier `0001-api-add-a-full_known_users-option-to-synchronization.patch` ajouté

- Statut changé de *En cours* à *Solution proposée*

- Patch proposed changé de *Non* à *Oui*

Paul Marillonnet a écrit :

Parce que c'est pour la synchro et que c'est la solution qui me semble encore la plus simple.

Voilà, une option `full_known_users` à cette API, qui :

- supporte de la pagination (avec un peu de code custom parce que dans notre cas d'usage on ne peut pas faire avec ce que DRF propose de façon native) ;
- ne reprend pas les identifiants inconnus, car il me semble que cette redondance n'a pas d'intérêt surtout si l'on tient compte de la pagination sur les informations d'utilisateurs connus renvoyées.

#9 - 26 octobre 2021 12:01 - Benjamin Dauvergne

- Statut changé de *Solution proposée* à *En cours*

- Assigné à mis à *Paul Marillonnet*

C'est finalement assez moche car il me semble que la pagination n'est pas tellement nécessaire finalement, pourvu que le client envoie une liste d'uuid inférieure à la taille maximum d'une page (c'est le client qui va paginer sa liste d'uuid en fait). Ça ne ressemble pas à une API qu'on voudrait supporter longtemps, je dirai de virer la partie pagination et de juste vérifier que la liste d'uuid ne dépasse pas une limite qu'on se fixe, comme 1000 (sans passer par `MAX_PAGE_SIZE`).

#10 - 26 octobre 2021 12:13 - Paul Marillonnet

Benjamin Dauvergne a écrit :

je dirai de virer la partie pagination et de juste vérifier que la liste d'uuid ne dépasse pas une limite qu'on se fixe, comme 1000 (sans passer par `MAX_PAGE_SIZE`).

Ok.

#11 - 27 octobre 2021 10:37 - Paul Marillonnet

- Fichier `0001-api-add-a-full_known_users-option-to-synchronization.patch` ajouté

- Statut changé de *En cours* à *Solution proposée*

Voilà, sans la pagination c'est vraiment plus succinct.

#12 - 27 octobre 2021 12:17 - Benjamin Dauvergne

- Statut changé de *Solution proposée* à *Solution validée*

Je vois que tu as choisi de ne pas faire couiner a2 quand ça dépasse 1000, soit. Je pense que sur cette base tu peux continuer ton code de synchro mais tu n'auras pas les nouveaux utilisateurs.

#13 - 02 novembre 2021 12:03 - Paul Marillonnet

Benjamin Dauvergne a écrit :

Je vois que tu as choisi de ne pas faire couiner a2 quand ça dépasse 1000, soit. Je pense que sur cette base tu peux continuer ton code de synchro mais tu n'auras pas les nouveaux utilisateurs.

Je regarde d'abord comment inclure [#57564](#) par dessus ce travail, ensuite se posera la question des nouveaux usagers.

#14 - 02 novembre 2021 12:04 - Paul Marillonnet

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

```
commit 42f2571feddf91f4253f513ebd31a23aa1568cdc
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date: Fri Oct 22 10:41:44 2021 +0200
```

```
api: add a full_known_users option to /synchronization/ endpoint (#57567)
```

#15 - 02 novembre 2021 12:06 - Paul Marillonnet

- Lié à *Development #58312*: inclure les nouveaux usagers au paramètre full de l'api de synchronisation (?) ajouté

#16 - 02 novembre 2021 19:16 - Frédéric Péters

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

Fichiers

0001-api-add-a-full_known_users-option-to-synchronization.patch	7,4 ko	25 octobre 2021	Paul Marillonnet
0001-api-add-a-full_known_users-option-to-synchronization.patch	4,07 ko	27 octobre 2021	Paul Marillonnet