w.c.s. - Development #48752

Gestion des accès aux API, association accès → rôles

23 novembre 2020 10:59 - Frédéric Péters

Statut:	Fermé	**Début:**	23 novembre 2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Frédéric Péters	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Pour le moment les accès aux API sont globaux et les permissions se font en fonction d'un paramètre email (ou NameID) passé aux requêtes, qui sert alors pour trouver un utilisateur; ça serait bien de pouvoir fixer une assocation entre un accès et des rôles, pour :

- 1/ ne plus imposer ce paramètre email;
- 2/ ne plus imposer la création d'un utilisateur bidon, qui s'est déjà parfois trouvé supprimé par erreur, et qui pourrait un jour être supprimé automatiquement par défaut de connexion;
- 3/ (last but not least) permettre de limiter les permissions accordées lorsqu'on accorde un accès aux API.

Techniquement, il s'agirait donc d'ajouter une sélection de rôles à l'objet créé via #48751 puis à modifier get_user_from_api_query_string() pour créer un objet "User" temporaire pour l'occasion.

Révisions associées

Révision 27756287 - 05 mai 2021 13:29 - Frédéric Péters

backoffice: add storage/UI to store/assign roles to API accesses (#48752)

Révision 674ab42b - 05 mai 2021 13:29 - Frédéric Péters

api: add roles-based access restrictions (#48752)

Historique

#1 - 24 novembre 2020 10:03 - Thomas Noël

Juste pour paraphraser, l'idée est donc qu'en l'absence de NameID ou email dans la query-string, ça soit le "orig=xxx" qui détermine le rôle avec lequel l'API est interrogée. Rôle qui est configuré dans la page de configuration de xxx, créée dans #48751.

#2 - 24 novembre 2020 10:25 - Frédéric Péters

> en l'absence de NameID ou email

Je voudrais l'affaire plus restrictive que ça, quand les rôles sont définis au niveau de l'accès API, ils prennent le pas sur NameID/email qui serait dans l'URL (il n'y aurait pas d'existant je serais même pour lever une erreur quand NameID/email se trouve dans la query string).

> qui détermine le rôle

Le ou les rôles.

#3 - 24 novembre 2020 10:55 - Benjamin Dauvergne

Thomas Noël a écrit :

> Juste pour paraphraser, l'idée est donc qu'en l'absence de NameID ou email dans la query-string, ça soit le "orig=xxx" qui détermine le rôle avec lequel l'API est interrogée. Rôle qui est configuré dans la page de configuration de xxx, créée dans #48751.

Fred veut créer deux types de comptes de services :

- les internes qui vont rester certainement dans site-options.cfg et qui ne souffrent d'aucune limitation
- les externes définis dans l'interface, où on pourrait même dire que poser des ~~restrictions~~ permissions via les rôles est fortement recommandé voir obligatoire et qui n'ont juste pas le droit d'utiliser la fonctionnalité de subrogation de l'identité

#4 - 17 avril 2021 15:04 - Frédéric Péters

- Assigné à mis à Frédéric Péters

#5 - 27 avril 2021 08:09 - Frédéric Péters

- Fichier 0002-api-add-roles-based-access-restrictions-48752.patch ajouté

- Fichier 0001-backoffice-add-storage-UI-to-store-assign-roles-to-A.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

0001 qui pose l'interface, c'est-à-dire dans l'édition d'un accès aux API un nouvel attribut "rôles", j'ai mis un texte d'explication "Apply access control related to these roles" qui peut sans doute être amélioré. 0001 déborde un peu sur d'autres fichiers parce que faire référence à des rôles à amener à devoir en stocker les identifiants et donc passer un paramètre include_id, ça a demandé la modification des signatures de méthodes export_XXX_to_xml et import_XXX_from_xml un peu partout, pour mettre **kwargs pour ne plus avoir y passer si jamais un nouveau paramètre se trouvait devoir être ajouté. (aussi par endroit en absorbant dans le kwargs le paramètre charset, dans l'idée d'en faciliter la suppression prochaine).

0002 pour le contrôle d'accès, une méthode get_as_api_user sur l'objet ApiAccess qui retourne un objet qui ressemble à un User (un attribut is_admin et une méthode get_roles); et si un objet ApiAccess est utilisé lors d'un appel à l'API et que des rôles ont été configurés sur celui-ci, c'est ce faux objet RestrictedApiUser qui est utilisé.

Dans les API de création de demande/fiche l'utilisateur, quand l'utilisateur n'était pas spécifié dans le payload, il pouvait être pris comme étant l'auteur de la requête, c'est contrôlé et bien sûr pas possible avec RestrictedApiUser.

Les tests couvrent l'API de récupération de fiches (/api/cards/test/list), de récupération d'infos de l'utilisateur (/api/user/, /api/user/forms) qui ne sont juste pas accessibles, et l'appel à un /jump/trigger/ d'un workflow.

#6 - 03 mai 2021 09:54 - Frédéric Péters

- Fichier 0002-api-add-roles-based-access-restrictions-48752.patch ajouté

- Fichier 0001-backoffice-add-storage-UI-to-store-assign-roles-to-A.patch ajouté

> (...) il pouvait être pris comme étant l'auteur de la requête, c'est contrôlé et bien sûr pas possible avec RestrictedApiUser.

c'était fait avec isinstance(...) là j'ai modifié pour une version avec un attribut (is_api_user) sur l'objet.

#7 - 05 mai 2021 01:35 - Thomas Noël

Pour 0001 :

- "Apply access control related to these roles" je suis tellement doué que je ne comprends pas l'anglais ici ; mais en français j'aurais dit "Rôles donnés par cet accès" ("Roles given with this access" ?)
- dans api_access.html on pourrait avoir un else sur le if api_access.get_roles, qui dirait « Rôles de l'utilisateur indiqué par email ou NameID » mais c'est du bavardage (c'est juste pour ne pas avoir à répondre à la question le jour où on va nous la poser... et je suis finalement pas sûr que mon idée clarifie les choses mais je ne l'efface pas allez)

Pour 0002 :

Je me pose la question de RestrictedApiUser qui est vraiment un User minimal. J'imagine d'éventuels conséquences par exemple sur ApiCardPage::submit avec usage de session_user_xxx dans le formulaire ou son workflow... mais ce n'est pas très grave, si on voit des problèmes de ce genre apparaître, on les corrigera. Autant garder la simplicité proposée ici pour l'instant.

Rien d'autre à dire, je m'attendais à un truc plus compliqué, c'est cool.

#8 - 05 mai 2021 08:20 - Frédéric Péters

> "Rôles donnés par cet accès"

Dans la forme j'aurais aimé rester similaire à "Restrict to anonymised data", i.e. on aurait "Limiter aux données anonymisées" et "Contrôler les accès selon ces rôles", genre, mais en fait le "Limiter aux données anonymisées" ça n'est déjà pas un texte d'aide c'est l'intitulé du champ, donc je cherche une cohérence qui n'existe déjà pas.

Je ne reste cependant pas bien convaincu de "Rôles donnés par cet accès" qui m'a l'air trop un raccourci. Mais comme je vais dire bof au point suivant, je fais ce changement ici.

> ... « Rôles de l'utilisateur indiqué par email ou NameID » ...

Plutôt bof donc, surtout parce que je ne pense pas qu'il faille faire la promotion de cette possibilité, et pour ne pas l'encourager, le mieux c'est de ne pas en mentionner l'existence, j'ai trouvé.

~~

(branche à jour avec le changement sur le texte d'aide)

#9 - 05 mai 2021 11:13 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Frédéric Péters a écrit :

> > ... « Rôles de l'utilisateur indiqué par email ou NameID » ...
>
> Plutôt bof donc, surtout parce que je ne pense pas qu'il faille faire la promotion de cette possibilité, et pour ne pas l'encourager, le mieux c'est de ne pas en mentionner l'existence, j'ai trouvé.

L'usage de email= a pourtant sa logique, cf #53736, permettre à un logiciel distant de savoir si son utilisateur local a le droit de jouer sur une demande. Mais oui pour ne pas mettre ce texte, moi même je le trouvais très bof.

#10 - 05 mai 2021 13:32 - Frédéric Péters

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

> L'usage de email= a pourtant sa logique, cf #53736, permettre à un logiciel distant de savoir si son utilisateur local a le droit de jouer sur une demande.

Tout à fait mais c'est plutôt rare d'avoir cette préoccupation qui correspond tout à fait à l'usage, et comme on n'avait pas d'autre possibilité habitude a été prise de l'utiliser avec un compte email avec plein de droits, tentative de casser cette habitude ici.

```
commit 674ab42b3a4de3bcc790e5d71f5904f4259d7d90
Author: Frédéric Péters <fpeters@entrouvert.com>
Date:   Sat Apr 17 14:40:28 2021 +0200

    api: add roles-based access restrictions (#48752)

commit 27756287a050c4d253a57974d8fc32886af6bf12
Author: Frédéric Péters <fpeters@entrouvert.com>
Date:   Sat Apr 17 13:08:16 2021 +0200

    backoffice: add storage/UI to store/assign roles to API accesses (#48752)
```

#11 - 05 mai 2021 23:16 - Frédéric Péters

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

Fichiers

0002-api-add-roles-based-access-restrictions-48752.patch	10,4 ko	27 avril 2021	Frédéric Péters
0001-backoffice-add-storage-UI-to-store-assign-roles-to-A.patch	12,3 ko	27 avril 2021	Frédéric Péters
0002-api-add-roles-based-access-restrictions-48752.patch	10,7 ko	03 mai 2021	Frédéric Péters
0001-backoffice-add-storage-UI-to-store-assign-roles-to-A.patch	12,3 ko	03 mai 2021	Frédéric Péters