Passerelle - Development #2861

système d'authentification

12 mai 2013 16:25 - Thomas Noël

Statut:	Fermé	**Début:**	12 mai 2013
Priorité:	Normal	**Echéance:**	07 octobre 2013
Assigné à:	Thomas Noël	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	

Description

Il faut penser à ajouter un système d'authentification au moins basique pour l'accès à certains webservices (exemple: intégration de données dans Solis).

Premières réponses faisables dans l'état actuel :

- filtrer certaines URLs au niveau apache
- ajouter un paramètre "&apikey=..." (ou équivalent) dans le modèle de données à restreindre

Plus tard, une réponse plus générique devra être étudiée. Sources d'inspiration :

- http://django-tastypie.readthedocs.org/en/latest/authentication.html et http://django-tastypie.readthedocs.org/en/latest/authorization.html
- http://django-rest-framework.org/tutorial/4-authentication-and-permissions.html

Historique

#1 - 04 juin 2013 15:46 - Thomas Noël

Cf Base

#2 - 11 juin 2013 15:47 - Thomas Noël

- Fichier passerelle-apiuser-start.patch ajouté

- Statut changé de Nouveau à En cours

Voici une première étape, ajoutant une application "base" dans Passerelle contenant :

- un modèle ApiUser tel que décrit dans Base
- un modèle BaseResource sur lequel se basent toutes les ressources proposées par Passerelle (i.e. datasource et repost pour l'instant)

Ce qui reste à faire :

- ajouter un middleware capable de détecter un apiuser (si signature, la vérifier et chercher l'apiuser ; si apikey, chercher l'apiuser)
- ajouter un contrôle de l'apiuser dans BaseResource.check_user() par exemple... voir comment proposer ça facilement dans les vues, idéalement de façon transparente, j'ai pas encore regardé comme faire joliment

Note annexe:

- la migration vers cette nouvelle architecture ajoute seulement des tables pour les manytomany, mais ça rend le syncdb inopérent : il va falloir passer par **South** pour migrer vers cette nouvelle version
- le modèle commun BaseResource devrait permettre de rendre le slug unique, et donc d'avoir juste des URL /slug (sans le prefix /data ou /repost ou autre) : le url.py chercherait la ressource et appelle ensuite le resolveur d'URL du type de ressource cible ? Mouaich... peut-être une fausse bonne idée
- peut-être ajouter un flag "public" pour permettre à une ressource d'être listée dans le frontoffice public même si elle est protégée (i.e. qu'elle possède au moins un "apiuser") -- sachant que par défaut, il faudra que seul un admin pourra voir toutes les ressources sur les pages frontoffice.

#3 - 11 juin 2013 15:51 - Thomas Noël

- Fichier passerelle-apiuser-start.patch ajouté

Oups... le patch complet...

#4 - 11 juin 2013 15:53 - Thomas Noël

- Fichier passerelle-apiuser-start.patch supprimé

#5 - 11 juin 2013 18:12 - Frédéric Péters

Je lisais il y a quelques jours http://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api#authentication

> By always using SSL, the authentication credentials can be simplified to a randomly generated access token that is delivered in the user name field of HTTP Basic Auth. The great thing about this is that it's completely browser explorable. […]

Et je trouve que ce serait en effet pas mal de gérer la situation username/key comme username/password, sur une authent HTTP. Non ?

#6 - 11 juin 2013 19:29 - Thomas Noël

Yep, on pourra l'ajouter au niveau du middleware à écrire, je pense quelque chose comme :

- si request.META['HTTP_AUTHORIZATION'] => unbase64 => user = chercher(origin, key)
- si signature dans query-string => user = chercher(origin) et vérifier le hmac avec sa clé
- si apikey => user = chercher(apikey)

#7 - 12 juin 2013 00:13 - Thomas Noël

- Fichier passerelle-apiuser-models-and-middleware.patch ajouté

Un patch plus complet, avec le middleware. Dans les vues on disposera alors d'un request.apiuser, qui sera un ApiUser ou None.

Le middleware détecte :

- les URLs signées avec ...&orig=username&algo=...&timestamp=...&nonce=...&signature=... si la clé de "username" est bien celle qui a produit la signature selon l'algo de http://doc.entrouvert.org/portail-citoyen/
- ou une &apikey=... dans la requête
- ou une authentification avec username et mot de passe correspondant à une clé de signature

#8 - 12 juin 2013 00:22 - Thomas Noël

- Fichier passerelle-apiuser-south.patch ajouté

L'ajout de "south" qui sera nécessaire pour faire ce qui précède.

#9 - 12 juin 2013 00:22 - Thomas Noël

- Fichier passerelle-apiuser-start.patch supprimé

#10 - 14 juin 2013 11:13 - Thomas Noël

- Statut changé de En cours à Résolu (à déployer)

Voilà, c'est poussé

#11 - 04 octobre 2013 13:37 - Thomas Noël

- Fichier 0001-access-control-through-apiuser.patch ajouté

- Echéance mis à 07 octobre 2013

- Statut changé de Résolu (à déployer) à En cours

- Assigné à mis à Thomas Noël

La suite, un peu passée aux oubliettes... la vérification des accès dans les vues.

J'en ai profité pour factoriser, rendre les vues un peu plus génériques, etc.

#12 - 04 octobre 2013 16:21 - Benjamin Dauvergne

Les méthodes comme get_from_slug() vont normalement sur l'objet Manager, mais je ne suis pas orthodoxe sur ce point.

J'aurai plutôt vu /templates/passerelle/base/ comme racine des nouveaux templates plutôt que /templates/base/ mais c'est personnel.

J'aurai bien vu une méthode get_for_user(user) sur le manager de l'objet BaseResource plutôt que de voir ce code uniquement dans ResourceIndexView.get_queryset.

Bon sinon les patchs qui contiennent plus de rouge que de vert, c'est trop bien.

#13 - 04 octobre 2013 18:27 - Thomas Noël

- Fichier 0001-access-control-through-apiuser.patch ajouté

Nouvelle version, qui intègre les remarques de Benji + l'adaptation de "queue" apparu entre temps.

#14 - 28 juin 2014 20:26 - Frédéric Péters

- Statut changé de En cours à Fermé

- Patch proposed mis à Non

Ça a été poussé depuis.

Fichiers

passerelle-apiuser-models-and-middleware.patch	8,88 ko	11 juin 2013	Thomas Noël
passerelle-apiuser-south.patch	11,3 ko	11 juin 2013	Thomas Noël
0001-access-control-through-apiuser.patch	26,4 ko	04 octobre 2013	Thomas Noël
0001-access-control-through-apiuser.patch	32,7 ko	04 octobre 2013	Thomas Noël