Fargo - Development #16192

ajouter dépôt de document en oauth2

05 mai 2017 14:56 - Jean-Baptiste Jaillet

Statut:	Fermé	**Début:**	05 mai 2017
Priorité:	Normal	**Echéance:**	
Assigné à:	Josué Kouka	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

Demandes liées:

Lié à Fargo - Development #14147: ajouter un accès oauth2 aux fichiers	**Fermé**	**29 novembre 2016 16 septembre 2017**
Lié à Fargo - Bug #16186: Destruction d'un objet Document : Document has no a...	**Fermé**	**05 mai 2017**
Lié à Fargo - Development #16842: Déléguer l'authentification des clients à a...	**Fermé**	**12 juin 2017**

Historique

#1 - 05 mai 2017 14:59 - Jean-Baptiste Jaillet

- Tracker changé de Bug à Development

http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=e5dd6f161bfcf1e879c6cfefeef7d67ce12a5c45

Commit du dépôt.

Entre autres remarques, vous pouvez enlever la remarque sur l'import de login_required qui ne sert à rien, je l'ai notée.

#2 - 05 mai 2017 15:00 - Jean-Baptiste Jaillet

- Lié à Development #14147: ajouter un accès oauth2 aux fichiers ajouté

#3 - 05 mai 2017 15:06 - Frédéric Péters

```
+            uri = request.session['redirect_uri'] + '?' + urllib.urlencode
```

...

#4 - 05 mai 2017 16:31 - Jean-Baptiste Jaillet

- Lié à Bug #16186: Destruction d'un objet Document : Document has no attribute document_file ajouté

#5 - 05 mai 2017 16:36 - Benjamin Dauvergne

Je relis plus sans tests.

#6 - 08 mai 2017 19:22 - Jean-Baptiste Jaillet

avec dernières modif plus test. Pour l'instant uniquement déroulement normal
http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=0533ead2db852f2d9f63fb484daefd00604de509

#7 - 10 mai 2017 09:16 - Benjamin Dauvergne

Dans mon souvenir HTTP Basic c'est login:password encodé en base64, comme je ne vois aucune b64decode, je suppose que ce n'est pas bon.

#8 - 10 mai 2017 10:38 - Jean-Baptiste Jaillet

Correction du base64 pour le HTTP Basic, prise en compte dans les tests :
http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=897fbef5b574eddec0049ac808b3ccf621801e27

Suite à la discussion dans #16223, ajout d'un modèle et modification de la gestion du put (c'est pour ça que je le laisse dans un commit séparé pour le moment): http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=5cb3664ddb14cc5e33a09143bc78a6ea4f98a0b6

Je remarque que les tests sur jenkins bug sur le app.post avec 5 arguments et je n'ai aucun souci en local. C'est sur la version de WebTestMixin qui pose souci ?

#9 - 13 mai 2017 16:39 - Frédéric Péters

> Je remarque que les tests sur jenkins bug sur le app.post avec 5 arguments et je n'ai aucun souci en local. C'est sur la version de WebTestMixin qui pose souci ?

...

As-tu testé avec tox, pour avoir un environnement similaire ?

Session :

```
tests/test_oauth2.py:92: TypeError
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> entering PDB >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
> /home/fred/src/eo/fargo/tests/test_oauth2.py(92)test_put_document()
-> resp = app.post(url, data, headers=headers, status=200)
(Pdb) p app
<django_webtest.DjangoTestApp object at 0x7f9a56e58e50>
(Pdb) p app.post
<bound method DjangoTestApp.post of <django_webtest.DjangoTestApp object at 0x7f9a56e58e50>>
(Pdb) p django_webtest
*** NameError: NameError("name 'django_webtest' is not defined",)
(Pdb) import django_webtest
(Pdb) p django_webtest
<module 'django_webtest' from '/home/fred/src/eo/venv/local/lib/python2.7/site-packages/django_webtest/__init_
_.pyc'>
```

vi du fichier en question.

```
    def post(self, url, **kwargs):
        extra_environ = kwargs.get('extra_environ')
        user = kwargs.pop('user', None)
        kwargs['extra_environ'] = self._update_environ(extra_environ, user)
        return super(DjangoTestApp, self).post(url, **kwargs)
```

En effet un seul paramètre positionnel (deux en comptant le self).

Monter dans la classe plus haut, .../webtest/app.py.

```
    def post(self, url, params='', headers=None, extra_environ=None,
             status=None, upload_files=None, expect_errors=False,
             content_type=None, xhr=False):
        """
        Do a POST request. Similar to :meth:`~webtest.TestApp.get`.

        :param params:
            Are put in the body of the request. If params is a
            iterator it will be urlencoded, if it is string it will not
            be encoded, but placed in the body directly.
```

"placed in the body directly", c'est ce qu'on veut.

```
-    resp = app.post(url, data, headers=headers, status=200)
+    resp = app.post(url, params=data, headers=headers, status=200)
```

#10 - 15 mai 2017 11:50 - Jean-Baptiste Jaillet

http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=6b652fc9d490cb5e92489546c04ac4939526399f

Modification des tests sur les paramètres de post et j'ai modifié le Content-disposition supposé d'arrivée.

#11 - 15 mai 2017 15:27 - Benjamin Dauvergne

Alors le parsing du header authorization, il vaudrait mieux le faire comme ceci:

```
    def authenticate(request):
        authorization = basic_auth = request.META.get('HTTP_AUTHORIZATION')
        if not authorization:
            return False
        splitted = authorization.split()
        if len(splitted) < 2:
            return False
```

```
        if splitted[0] != 'Basic':
            return False
        try:
            decoded = base64.b64decode(splitted[1])
        except ValueError:
            return False
        credentials = decoded.split(':', 1)
        if len(credentials) < 2:
            return False
        client_id, client_secret = credentials
        try:
            return OAuth2Client.objects.get(client_id=client_id, client_secret=client_secret)
        except OAuth2Client.DoesNotExist:
            return False
```

Je serai plus rassuré, et quand c'est False, renvoyer une erreur 401 me semble suffisant.

Dans les test plutôt utiliser l'API webtest de base pour faire de l'authentification:
http://docs.pylonsproject.org/projects/webtest/en/latest/testapp.html#modifying-the-environment-simulating-authentication

Dans la mesure ou le #16186 a été poussé, il faut décommenter la ligne qui fait le delete.

Il faudrait gérer le cas où on revient sur l'URL et le document a été supprimé (i.e. quand Document.objects.get() fait un raise DoesNotExist dans OAuth2AuthorizePutView.get_context_data(), par exemple après un cancel si la personne fait Back, dans ce cas il faut juste considérer qu'il y a un cancel implicite).

#12 - 15 mai 2017 17:34 - Jean-Baptiste Jaillet

J'ai fait les modifs mais ce n'est pas encore poussé.
Pour la dernière partie de tes commentaires, est ce que tu veux dire que si la personne clique sur précédent dans son navigateur, on fait la même chose que cancel? (et si oui, c'est un truc gérable avec du js ?)

Pour le commentaire sur le delete() à decommenter, il y a le dernier patch sur ma branche qui change de stratégie pour le document temporaire, et donc cette partie là est changée. Mais au moins le delete() de document est bon maintenant.

Remarque en lien avec le ticket pour récupérer un fichier, est ce qu'on fait pareil pour le passage de token (Bearer), un truc dans l'idée de la fonction authenticate ?

#13 - 17 mai 2017 13:49 - Benjamin Dauvergne

Jean-Baptiste Jaillet a écrit :

> J'ai fait les modifs mais ce n'est pas encore poussé.
> Pour la dernière partie de tes commentaires, est ce que tu veux dire que si la personne clique sur précédent dans son navigateur, on fait la même chose que cancel? (et si oui, c'est un truc gérable avec du js ?)

En fait il faut simplement que tu gères les cas d'accès multiples à l'URL de confirmation, il y a 3 cas:

- l'usager n'a rien choisi au dernier accès: rien à faire on réaffiche la même page
- l'usager a fait cancel au dernier accès: l'objet Document n'existe plus, ce que je dis c'est qu'il faut gérer ça proprement en considérant qu'un cancel à du avoir eu lieu et simplement retourner sur redirect_uri (ou affiche une page intermédiaire qui dit que le document n'est plus là, certainement à cause d'un cancel, et fournir un lien vers callback_url)
- l'usager a accepté au dernière accès: l'objet Document est déjà lié à l'utilisateur, je propose là aussi de ne rien faire à part dire à l'utilisateur que le document est déjà dans son porte-doc, et de proposer un lien vers redirect_uri.

  > Pour le commentaire sur le delete() à decommenter, il y a le dernier patch sur ma branche qui change de stratégie pour le document temporaire, et donc cette partie là est changée. Mais au moins le delete() de document est bon maintenant.

Ok avec ça, concernant mes remarques plus haut, elles sont toujours valide mais il faut les appliquer à OAuth2TempFile et pas Document.

> Remarque en lien avec le ticket pour récupérer un fichier, est ce qu'on fait pareil pour le passage de token (Bearer), un truc dans l'idée de la fonction authenticate ?

Oui, le code est juste un peu plus simple puisqu'il n'y a pas de décodage du payload après "Bearer" comme pour "Basic".

#14 - 18 mai 2017 18:39 - Jean-Baptiste Jaillet

Mise à jour des dernières remarques, et j'ai aussi intégré la création du nouveau fichier temporaire dans le commit de put.
Mise à jour des tests et traductions.

http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=8dd9820224ed3661c7715879914d2260770d4c66

#15 - 19 mai 2017 10:25 - Benjamin Dauvergne

Toujours pareil, tu supposes que tout va bien se passer, qu'on t'envoie les bonnes données, etc.. il faut être plus défensif, ça peut être bien que tu fasses des tests qui envoie des données pourries aussi pour bien voir ce qui se passe.

```
+    filename = request.META['HTTP_CONTENT_DISPOSITION'].replace(' ', '').split(';')[1]
+    filename = filename.replace('filename=', '').replace('"', '')
```

Que se passera-t-il si HTTP_CONTENT_DISPOSITION n'est pas là ? ou si son contenu est pourrave ? Il faut détecter tout ça et renvoyer des erreurs 400 intelligibles (Missing content-disposition header, Invalid content-disposition header, etc..).

D'après une question[1] stackoverflow tu peux utiliser cgi.parse_header pour t'aider. Mais ça donnera ça:

```
>>> cgi.parse_header('''attachment; filename*= UTF-8''%e2%82%ac%20rates''')
('attachment', {'filename*': "UTF-8''%e2%82%ac%20rates"})
```

Il faudra encore décoder la chaîne dans le cas de filename*, valider que l'encoding est connu, décoder le reste (on ignore l'indication de pays entre les deux simple-quote). On préférera toujours la valeur extraite de filename* à celle dans filename[2]

Comme pour l'authentification je te conseille de créer une fonction filename, error = get_filename(request) pour gérer tout ça tranquillement:

```
def get_filename(request):
    if 'HTTP_CONTENT_DISPOSITION' not in request.META:
        return None, 'Missing Content-Disposition header'

    etc...

    return filename, None
```

[1] http://stackoverflow.com/questions/8035900/how-to-get-filename-from-content-disposition-in-headers

[2] https://tools.ietf.org/html/rfc6266#section-4.3

Ces chaînes ne sont pas localisées:

```
+            context['error_message'] = 'The document has not been uploaded'
+            context['error_message'] = 'This document is already in your portfolio'
```

Pouruqoi un deuxième template ? Un {% if %} dans le premier template ça ne suffit pas ?

```
+    def render_to_response(self, context, **kwargs):
+        if 'error_message' in context:
+            self.template_name = 'oauth2/confirm_exception.html'
+
+        return super(OAuth2AuthorizePutView, self).render_to_response(context, **kwargs)
```

Idem, pas de redirect_uri, boum.

```
+        request.session['redirect_uri'] = request.GET['redirect_uri']
```

Ici le code n'est pas clair:

```
+    def get_context_data(self, **kwargs):
+        context = super(OAuth2AuthorizePutView, self).get_context_data(**kwargs)
+        try:
+            oauth2_document = OAuth2TempFile.objects.get(pk=kwargs['pk'])
+
+            user_document = UserDocument.objects.get(user=self.request.user,
+                                                     document=oauth2_document.document)
+        except OAuth2TempFile.DoesNotExist:
+            context['error_message'] = 'The document has not been uploaded'
+            context['redirect_uri'] = self.request.GET['redirect_uri']
+        except UserDocument.DoesNotExist:
+            context['filename'] = oauth2_document.filename
+
+        else:
+            context['error_message'] = 'This document is already in your portfolio'
+            context['redirect_uri'] = self.request.GET['redirect_uri']
+
+        return context
```

Sépare mieux le chemin passant (normal pas d'erreur) des chemins non passants (un truc ne va pas), là on a pas passant, passant, pas passant, il vaudrait mieux grouper les cas non-passant au début, avec des return directs (ça évite de se poser la question quand on lit du code si il se passe un truc plus loin ou pas). Mets plus de commentaires surtout sur les cas non-passant, en général le cas passant est évident dans une fonction courte.

```
si OAuth2Document n'existe pas:
  # commentaire
  on pose des variables dans context
  return

si UserDocument existe:
  # commentaire
  on pose des variables dans context
  return

ok tout va bien, on pose des variables dans context
return
```

#16 - 21 mai 2017 18:54 - Jean-Baptiste Jaillet

Poussé dans http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=7b3484bb8a255730750f189094437c29529c1caf

Pour l'histoire des chaînes localisées, je mets ces valeurs dans error message et d'après la doc s'il y a un {% trans var %} il va chercher la première chaîne dispo.
J'ai bien mis dans le .po la bonne ligne du template où error_message est affichée.

#17 - 21 mai 2017 23:39 - Thomas Noël

Il y a un nouveau modèle, il doit y avoir une nouvelle migration.

Par ailleurs, déjà dit je-sais-plus-où, c'est un peu du détail, mais on pousse les .po séparément (en général on les pousse pas avec le patch, on fait une traduction qlq temps avant la release/tag). Ca évite de se casser les .po les uns les autres, parce que sinon c'est un fichier où on a le plus de chance de bosser à plusieurs, et ça marche mal avec les vcs. Bref, pas la peine de t'occuper du .po quand tu codes :)

#18 - 22 mai 2017 17:52 - Jean-Baptiste Jaillet

C'est noté pour les po je vais les enlever du commit.
Par contre pour les migrations j'ai vérifié elles sont bien dans le commit (pour le nouveau modèle de fichier temporaire).
Les autres modèles sont dans l'autre commit de création de l'api + ajout du get document. Je pense que c'était sur ça ta remarque?

#19 - 22 mai 2017 18:42 - Jean-Baptiste Jaillet

- Fichier 0002-oauth2-add-put-document-method.patch ajouté

- Patch proposed changé de Non à Oui

Ok j'ai viré les .po du commit.
J'ai un souci avec le push --force qui met pas à jour la branche, j'ai essayé de delete et recréer mais ça ne change rien.
Je mets un patch en attendant de régler tout ça, pour qu'on puisse relire quand même.

#20 - 22 mai 2017 18:46 - Jean-Baptiste Jaillet

J'avais un merge qui traînait...
Du coup http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=eb34d16ed55689a026580dac2e9d69ac6279b8ea.

#21 - 22 mai 2017 19:54 - Thomas Noël

Jean-Baptiste Jaillet a écrit :

> Par contre pour les migrations j'ai vérifié elles sont bien dans le commit (pour le nouveau modèle de fichier temporaire).

Je regarde http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=eb34d16ed55689a026580dac2e9d69ac6279b8ea

Tu ajoutes un modèle : class OAuth2TempFile(models.Model):

Mais tu **modifies** une migration existante. Ca ne se fait normalement jamais ; on le fait parfois à Entr'ouvert quand la modification d'un modèle n'engendre aucune modif de structure en base (par exemple, changer un label ou un help_text), c'est le seul et unique cas.

Donc là, il faut **ajouter** une migration (enfin faut faire confiance à makemigrations et c'est tout, y'a rien à coder quoi)

#22 - 24 mai 2017 12:38 - Jean-Baptiste Jaillet

http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=5f39cc787b2fe66709246d99a9f88edb139ea090

C'est mis en deux migrations.

#23 - 24 mai 2017 14:38 - Thomas Noël

Sur la forme, ces détails qui comptent dans notre façon de coder proprement :

```
    url(r'put-document/(?P<pk>\w+)/authorize', login_required(OAuth2AuthorizePutView.as_view()), name='oauth2-put-document-authorize')
```

ligne de 130 caractères, trop longue (pep8 c'est 79 caractères, chez EO on accepte éventuellement jusqu'à 100 quand ça évite du bordel, c'est pas le cas ici)

Dans le views.py : # -*- coding: utf-8 -*- est inutile. Généralement on n'a pas besoin d'utf-8 dans notre code, l'ascii suffit (on écrit tout en américain, les traductions gère l'utf8)

Tu mets souvent une espace avant ":", ça se fait pas en américanie. Genre 'unknown encoding : ...' → 'unkown encoding: ...'

Un peu trop de saut de lignes dans les fonctions à mon goût (bon là c'est parfois une question de goût, mais par exemple un saut de ligne en fin de bloc avant un "else:", je trouve ça inutile/moche. (oui, chui relou)

Des lignes vide en bas des fichiers (ça, même "git show" me les montre en rouge)

Dans les tests, quand on utilise des domaines, c'est example.net ou example.org (RFC 2606, https://en.wikipedia.org/wiki/example.net)

Sur le fond j'ai un peu trop mal au crane là, j'imagine qu'il faut d'abord valider le premier patch #14147

#24 - 24 mai 2017 15:52 - Jean-Baptiste Jaillet

Ok j'ai modifié ça : http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=314c2e1eea78959fa00c4c0ac69421572cdebbc3

#25 - 26 mai 2017 09:25 - Jean-Baptiste Jaillet

Mise à jour : http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=82474b8e34e403376b24b5b0cab54492a926b9bd

#26 - 29 mai 2017 11:27 - Jean-Baptiste Jaillet

Mise à jour du commit par rapport à la modification des patterns dans #14147 :
http://repos.entrouvert.org/fargo.git/commit/?h=wip/oauth2&id=f1e78c91278721757fbc51e72f77f67540a14549

#27 - 05 juillet 2017 17:46 - Josué Kouka

- Lié à Development #16842: Déléguer l'authentification des clients à authentic. ajouté

#28 - 28 août 2017 20:23 - Frédéric Péters

- Assigné à changé de Jean-Baptiste Jaillet à Josué Kouka

#29 - 08 novembre 2017 10:23 - Josué Kouka

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

#30 - 06 mars 2018 12:13 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0002-oauth2-add-put-document-method.patch	13,3 ko	22 mai 2017	Jean-Baptiste Jaillet