Authentic 2 - Development #26022

photo de profil / avatar dans le profil

03 septembre 2018 09:56 - Frédéric Péters

Statut:	Fermé	**Début:**	03 septembre 2018
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

Par email,

> (...) l'idée d'avoir un nouveau type d'attribut pour uploader un fichier, stocker ce fichier et servir l'URL publique du fichier dans les attributs de SSO.

Demandes liées:

Lié à Authentic 2 - Development #26251: avatar: cadrage côté client de l'imag...	**En cours**	**08 septembre 2018**
Lié à Authentic 2 - Support #26250: avatar: test côté client de la dimension ...	**Nouveau**	**08 septembre 2018**
Lié à Authentic 2 - Support #26249: avatar : validation côté client de la tai...	**Fermé**	**08 septembre 2018**

Révisions associées

Révision a5d652ce - 30 octobre 2018 10:23 - Paul Marillonnet

support avatar picture in user profile (#26022)

Historique

#2 - 03 septembre 2018 10:35 - Paul Marillonnet

- Assigné à changé de Benjamin Dauvergne à Paul Marillonnet

#3 - 03 septembre 2018 14:25 - Paul Marillonnet

Il ne me semble pas pertinent de modifier le modèle custom_user d'A2.

Je verrais ça :

- définir une nouvelle entrée dans le dico DEFAULT_ATTRIBUTE_KIND, dont l'entrée 'field_class' serait un django.forms.ImageField.
- sérialisation et desérialisation simplement un dump du png ou jpg.
- une méthode qui génère une "remote_photo_url" qui serait servie aux SP une fois l'attribut activé
- une vue DRF pour servir l'image en elle même à partir de cette URL, sans doute une URL publique
- l'intégration BO : possibilité pour l'admin fonctionnel d'activer le support des images de profil, dans l'onglet 'Système' -> 'Profil Usager'
- l'intégration FO : possibilité pour l'usager de charger une image dans la gestion du compte, "Éditer les données du compte".

Je me plante ? J'oublie quelque chose ?

#4 - 03 septembre 2018 15:21 - Paul Marillonnet

La gestion des attributs dans l'onglet "Système" -> "Profil Usager" du BO semble complètement cloisonnée à hobo.
J'ai défini un nouvel attribut "Photo Avatar" dans l'interface /admin/ d'A2.
Je suis en train de creuser, voir comment "répercuter", d'une manière ou d'une autre, ces changements côté hobo, pour l'affichage dans le profil usager.

Edit: Je pensais que les types d'attributs pour ce profil usager étaient (tout comme les attributs en eux-mêmes), approvisionnés depuis A2, mais on dirait que non. Dans hobo.profile.models :

```
class AttributeDefinition(models.Model):
    % [...]
    kind = models.CharField(max_length=16, verbose_name=_('kind'), default='string',
                            choices=(('string', _('String')), ('boolean', _('Boolean'))))
```

Est-ce que je loupe une commande, de l'agent hobo, par exemple, pour déclencher un tel approvisionnement ?

#5 - 03 septembre 2018 15:47 - Paul Marillonnet

- Fichier screenshot_avatar_photo00.png ajouté

j'ai écrit :

- l'intégration FO : possibilité pour l'usager de charger une image dans la gestion du compte, "Éditer les données du compte".

Simplement quelque chose comme ça ? (cf capture du début de proto en PJ)
Et donc, il faut faire aussi en sorte d'afficher cette image dans la page principale de gestion du compte de l'usager.

#6 - 03 septembre 2018 16:16 - Mikaël Ates (de retour le 29 avril)

Je ne connais pas bien le cas d'usage mais à lire "l'URL publique du fichier" il semble que la photo sera posée en ligne sans restriction d'accès. Dans ce cas l'identifiant de la photo dans l'URL est une valeur opaque ? Pourquoi ne pas la servir directement au sso ou via l'API ?

#7 - 03 septembre 2018 19:56 - Paul Marillonnet

Je commence à comprendre un peu le code de gestion des attributs et valeurs d'attributs dans le profil utilisateur extensible d'A2.
Pour une raison qui m'échappe encore, le fichier envoyé ne se retrouve pas dans files à l'exécution de django.widgets.ClearableFileInput::value_from_datadict (qui est le widget de gestion des FileField).

Je continue de chercher.

#8 - 03 septembre 2018 20:36 - Paul Marillonnet

Mikaël Ates a écrit :

> Je ne connais pas bien le cas d'usage mais à lire "l'URL publique du fichier" il semble que la photo sera posée en ligne sans restriction d'accès. Dans ce cas l'identifiant de la photo dans l'URL est une valeur opaque ? Pourquoi ne pas la servir directement au sso ou via l'API ?

L'URL publique, je crois, ferait partie de l'API. Et oui donc, une URL opaque.
(et dans ce cas, il faudrait rajouter un texte explicatif pour l'usager, qu'il comprenne que cette image servie serait accessible aux fournisseurs de services).

Une URL restreinte via l'API, ça risque de limiter les cas d'usage, non ? (par exemple le SP qui provisionne un compte local, je suis pas certain que ça marche encore).

Et via le SSO, ce serait un dump du fichier image en base64 ? Je ne sais pas si ça complexifie ou non la chose (côté SP notamment).

#9 - 03 septembre 2018 22:19 - Frédéric Péters

> Et via le SSO, ce serait un dump du fichier image en base64 ? Je ne sais pas si ça complexifie ou non la chose (côté SP notamment).

Dans le propos de Benjamin repris en description : "servir l'URL publique du fichier dans les attributs de SSO".

#10 - 04 septembre 2018 15:42 - Benjamin Dauvergne

Dans mon idée on stockait le fichier dans /var/lib/a2-multitenant/tenants/<host>/media/avatars/, on publiait directement ce répertoire via nginx et on servait simplement des chemins vers celui-ci, en base on stocke juste le nom du fichier sur le disque.

#11 - 04 septembre 2018 15:43 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

> Dans mon idée on stockait le fichier dans /var/lib/a2-multitenant/tenants/<host>/media/avatars/, on publiait directement ce répertoire via nginx et on servait simplement des chemins vers celui-ci, en base on stocke juste le nom du fichier sur le disque.

Et je propose de générer un uuid comme nom de fichier au cas où, pas la peine de diffuser plus d'informations publiques que nécessaire.

#12 - 04 septembre 2018 15:44 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

> Benjamin Dauvergne a écrit :
>
> > Dans mon idée on stockait le fichier dans /var/lib/a2-multitenant/tenants/<host>/media/avatars/, on publiait directement ce répertoire via nginx et on servait simplement des chemins vers celui-ci, en base on stocke juste le nom du fichier sur le disque.

> Et je propose de générer un uuid comme nom de fichier au cas où, pas la peine de diffuser plus d'informations publiques que nécessaire.

Penser à stocker les fichiers dans avatars/<uuid>[0:4]/<uuid>.jpeg pour éviter d'avoir 80 000 fichiers dans le même répertoire.

#13 - 04 septembre 2018 16:28 - Paul Marillonnet

- Fichier 0001-WIP-support-avatar-picture-in-user-profile-26022.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Je joins pour infos un début d'implémentation côté interface de gestion du compte en front office.

Ok, merci Benjamin et Frédéric pour vos remarques. Je vais me creuser la tête pour le stockage du fichier, la génération de l'uuid et l'intégration d'un API publique.

#14 - 04 septembre 2018 16:29 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

- Patch proposed changé de Oui à Non

#15 - 04 septembre 2018 20:34 - Paul Marillonnet

- Fichier 0001-WIP-support-avatar-picture-in-user-profile-26022.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Paul Marillonnet a écrit :

> Je vais me creuser la tête pour le stockage du fichier, la génération de l'uuid et l'intégration d'un API publique.

En patch WIP joint un début de pseudo-code pour le stockage.
Je reprends demain.

#16 - 04 septembre 2018 20:34 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

- Patch proposed changé de Oui à Non

...

#17 - 05 septembre 2018 17:54 - Paul Marillonnet

- Fichier 0001-WIP-support-avatar-picture-in-user-profile-26022.patch ajouté

- Fichier account_avatar.png ajouté

Une capture, et un bout de code, pour avoir une idée de la direction que prend l'affaire :)

Pour l'instant le code ne fait que reconnaître qu'il s'agit d'une image, et l'affiche.

Encore rien niveau API, juste une URL publique.
Rien non plus pour l'instant au niveau validation de la taille de l'image, ni de l'ajustement des templates gérer l'affichage de l'avatar.

Je suis sur le coup.

#18 - 06 septembre 2018 14:59 - Paul Marillonnet

Ce qui me paraît le plus crucial à implémenter en premier maintenant :
- le formulaire d'édition du profil retrouve l'image d'avatar courante
- si l'usager ne touche pas à ce champ d'image, alors on la garde telle quelle
- le formulaire d'édition permet à l'usager de supprimer son image d'avatar, sans la remplacer par une nouvelle

Je ne suis pas non plus pour garder, dans MEDIA_ROOT/avatars/<user.uuid>/, toutes les images d'avatar précédentes.
Je voudrais que ce dossier ne contienne qu'un seul fichier au maximum.

Edit: Pour l'instant je n'arrive pas à tirer parti de django.forms.widget.ClearableFileInput, qui pourtant m'aiderait bien pour y parvenir.
Je crois que je vais commencer par chercher de ce côté-là.

Edit2: La version 1.11 de Django casse la rétrocompatibilité de la gestion des templates dans les widgets de champs de fichier. Je regarde comment faire tourner ça à la fois en 1.8 (puisque ça doit passer en recettes dans 2 semaines sur le SaaSv1) et en 1.11.

Edit3: En fait, après lecture du code source de Django 1.8 et 1.11 pour le rendu des widgets, c'est plus clair pour moi.

#19 - 08 septembre 2018 13:11 - Paul Marillonnet

- Lié à Development #26251: avatar: cadrage côté client de l'image chargée ajouté

#20 - 08 septembre 2018 13:11 - Paul Marillonnet

- Lié à Support #26250: avatar: test côté client de la dimension de l'image chargée ajouté

#21 - 08 septembre 2018 13:11 - Paul Marillonnet

- Lié à Support #26249: avatar : validation côté client de la taille de l'image ajouté

#22 - 11 septembre 2018 17:19 - Paul Marillonnet

- Fichier 0001-WIP-support-avatar-picture-in-user-profile-26022.patch ajouté

Ça avance. Je pose un patch WIP là, et je continue de me creuser la tête sur l'intégration dans l'API et dans les attributs envoyés au SSO.

#23 - 12 septembre 2018 17:28 - Paul Marillonnet

- Fichier 0001-WIP-support-avatar-picture-in-user-profile-26022.patch ajouté

Côtés API et attributs de SSO c'est bon.

Je vais écrire les tests, et écrire le gabarit d'affichage du widget qui déprécie le HTML inline dans Django 1.8, et tirer parti de sorl.thumbnail pour l'affichage de l'image, comme le conseille Frédéric dans #26250.

#24 - 17 septembre 2018 19:04 - Paul Marillonnet

- Fichier 0001-WIP-support-avatar-picture-in-user-profile-26022.patch ajouté

Le patch WIP avec l'intégration sorl-thumbnail.

Les quelques dernières pistes avant relecture :

- vérifier encore qu'il y a bien un bogue dans webtest (version 2.0.30) lors de l'annuliation (`form.submit(cancel)`) d'un formulaire encodé en multipart/form-data. Le code de webtest.forms.submit_fields semble (tout comme celui de webtest.app.encode_multipart) attendre une valeur de champ non nulle pour retrouver ce champ avant encodage du formulaire. Mais il teste l'égalité de la valeur contre webtest.forms.Submit.value_if_submitted() qui vaut toujours None.
Cf en particulier ce code de webtest.forms :

```
class Form(object):
    # [...]
    def submit_fields(self, name=None, index=None, submit_value=None):
        for name, field in self.field_order:
            # [...]
            if submit_name is not None and name == submit_name:
                # [...]
                if submit_value is not None and \
                   field.value_if_submitted() == submit_value:
                    submit.append((field.pos, name, field.value_if_submitted()))
```

- la configuration de la hauteur, largeur, et du recadrage de thumbnail (pour l'instant fixé en brut dans le template)

- sans doute virer les `eval` servant aux keyword arguments du sérialisateur des images (peut-être simplement des flags {'uuid_as_kwarg': True, 'label_as_kwarg': True}).

- la compatibilité django 1.11, en particulier le template du widget d'affichage de l'image qui se fait dans un fichier à part, et non plus en inline dans la définition de la classe comme c'est le cas en 1.8.

Edit:
- utiliser un dossier temporaire (fixture tempfile) pour le MEDIA_ROOT lors de l'exécution des tests.

#25 - 18 septembre 2018 11:35 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Paul Marillonnet a écrit :

> - vérifier encore qu'il y a bien un bogue dans webtest (version 2.0.30) lors de l'annuliation (`form.submit(cancel)`) d'un formulaire encodé en multipart/form-data. Le code de webtest.forms.submit_fields semble (tout comme celui de webtest.app.encode_multipart) attendre une valeur de champ non nulle pour retrouver ce champ avant encodage du formulaire. Mais il teste l'égalité de la valeur contre webtest.forms.Submit.value_if_submitted() qui vaut toujours None.

Je reste là-dessus, je ne vois pas moyen de faire fonctionner ces tests avec un formulaire en multipart/form-data (à moins de patcher python-webtest).

> - la configuration de la hauteur, largeur, et du recadrage de thumbnail (pour l'instant fixé en brut dans le template)

Fait.

> - sans doute virer les `eval` servant aux keyword arguments du sérialisateur des images (peut-être simplement des flags {'uuid_as_kwarg': True, 'label_as_kwarg': True}).

Je veux bien l'avis de relecteur(s) là-dessus svp (cf les modifs dans models.Attribute.set_value).
Les eval m'ont l'air bien saugrenu, mais je ne trouve rien d'autre qui tienne la route.

> - la compatibilité django 1.11, en particulier le template du widget d'affichage de l'image qui se fait dans un fichier à part, et non plus en inline dans la définition de la classe comme c'est le cas en 1.8.

Fait.

> Edit:
> - utiliser un dossier temporaire (fixture tempfile) pour le MEDIA_ROOT lors de l'exécution des tests.

Fait.

#26 - 18 septembre 2018 11:50 - Frédéric Péters

```
                {% with request.is_secure|yesno:"https://,http://"|add:request.get_host|add:attribute.values.0
 as im_url %}
                  {% thumbnail im_url img_dimensions crop=img_cropping as thumb_im %}
```

Non, il doit être possible de faire {% thumbnail attribute.values.0 as im %}. (thumbnail accepte différents types d'objets, genre File)

```
    {% if form.is_multipart %}
```

(2×) Je dirais inutile cette partie, que le formulaire soit toujours soumis en multipart/form-data ne heurtera personne.

```
def _store_image(in_memory_image, owner_uuid, attr_label):
```

tout ça m'a l'air bien long pour écrire un fichier.

```
    img_media_dir = '{label}/{oid}/'.format(
            oid=owner_uuid,
            label=attr_label)
```

Le commentaire de Benjamin disait avatars/<uuid>[0:4]/<uuid>.jpeg, pour éviter un répertoire avec autant d'entrées que d'utilisateurs. Ce que tu crées ici.

#27 - 18 septembre 2018 16:02 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Frédéric Péters a écrit :

> Non, il doit être possible de faire {% thumbnail attribute.values.0 as im %}. (thumbnail accepte différents types d'objets, genre File)

Je manipule ici des URI relatives d'image (sans le nom d'hôte, dans leur forme /<MEDIA_URL>/<label d'attribut/<uuid[0:4]>/<uuid>.ext), et ce format n'est pas pris en compte par sorl-thumbnail. Ce module ne prend en charge que les URL absolues, et les objets de fichiers (les InMemoryFile par exemple) [1].

Est-ce qu'on pourrait se contenter d'une version intermédiaire, comme dans le nouveau patch ici ?

(Et donc aussi garder dans les arguments de dimensions et de cadrage directement dans la balise de template ? Pour ces arguments je vois pas de solution plus simple, mais je passe peut-être à côté de quelque chose ?)

Dans l'idée trouve ça plus simple que :
- désérialiser les images dans un InMemoryFile simplement pour le rendu du template
- utiliser des URLs absolues dans des parties du code qui je crois n'ont pas à manipuler le nom d'hôte du serveur

Je me plante ?

> (2×) Je dirais inutile cette partie, que le formulaire soit toujours soumis en multipart/form-data ne heurtera personne.

Corrigé, merci.

> tout ça m'a l'air bien long pour écrire un fichier.

J'ai réduit un peu la fonction en gardant ce qui me paraissant nécessaire.

> Le commentaire de Benjamin disait avatars/<uuid>[0:4]/<uuid>.jpeg, pour éviter un répertoire avec autant d'entrées que d'utilisateurs. Ce que tu crées ici.

J'avais mal lu, c'est corrigé. Merci.

Corrigé aussi au passage la rétrocompatibilité cassée de l'interface custom_user.models.User.to_json()

[1] Cf ce bout de code sorl-thumbnail dans sa branche master actuelle, dans sorl.thumbnail.images :

```
url_pat = re.compile(r'^(https?|ftp):\/\/')
# [...]
class ImageFile(BaseImageFile):
    # [...]
    def __init__(self, file_, storage=None):
    # [...]
        # Support for relative protocol urls
        if self.name.startswith('//'):
            self.name = 'http:' + self.name

        # figure out storage
        if storage is not None:
            self.storage = storage
        elif hasattr(file_, 'storage'):
            self.storage = file_.storage
        elif url_pat.match(self.name):
            self.storage = UrlStorage()
        else:
            self.storage = default_storage
    # [...]
```

(https://github.com/jazzband/sorl-thumbnail/blob/master/sorl/thumbnail/images.py#L83)

#28 - 18 septembre 2018 16:06 - Frédéric Péters

> Je me plante ?

cf https://git.entrouvert.org/combo.git/tree/combo/apps/assets/templatetags/assets.py, fais un templatetag qui accepte le AttributeValue, et retourne l'URL.

#29 - 18 septembre 2018 17:50 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Frédéric Péters a écrit :

> cf https://git.entrouvert.org/combo.git/tree/combo/apps/assets/templatetags/assets.py, fais un templatetag qui accepte le AttributeValue, et retourne l'URL.

Merci, j'étais passé à côté. J'ai fait une balise d'attribution (assignment_tag) pour respecter la compatibilité Django 1.8 [1] même si c'est jugé obsolète en 1.11.

Je soupçonne le rafraichissement de cache sorl-thumbnail de ne se baser que sur un changement d'URL de fichier.
En tout cas il y avait clairement un problème de cache avec le format de nom d'image actuel, et donc c'est réglé maintenant que je suis passé d'un format de nom d'image :

```
/<label d'attribut>/<uuid[0:4]>/<uuid>.<format de l'image>
```

à

```
/<label d'attribut>/<uuid[0:4]>/<uuid><empreinte md5 de l'image en hexadécimal>.<format de l'image>
```

[1] Cf https://docs.djangoproject.com/fr/1.11/howto/custom-template-tags/#assignment-tags
"Obsolète depuis la version 1.9: simple_tag est dorénavant capable de stocker son résultat dans une variable de gabarit et devrait être privilégié."

#30 - 19 septembre 2018 09:02 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Il restait dans mes sources un fichier src/authentic2/templatetags/__init__.pyc, sans le __init__.py indexé par git.
C'est corrigé ici : ajout du __init__.py.

#31 - 19 septembre 2018 09:23 - Frédéric Péters

(c'est vraiment juste mon confort mais tu peux taper des images plus petites pour les tests ? (genre dans gimp dessine un carré bleu et un carré rouge et hop))

```
+                    {% get_image_url attribute.values.0 request as img_url %}
+                    {% thumbnail img_url img_dimensions crop=img_cropping as thumb_im %}
```

Je pensais moi vraiment à un {% thumbnail attribute.values.0 as thumb_im %}, pas de construction temporaire inutile d'URL.

```
+                      <img class={{ attribute.attribute.name }} src="{{ thumb_im.url }}">
```

(il faut des guillemets autour de la classe, et curieux tu as un espace insécable derrière le premier {{. (je te suggérerais de configurer ton éditeur pour les rendre visibles)

Je reste sur mon avis que _store_image peut être réduit, davantage encore. (et je dirais aussi qu'on se fout du hash de l'image) (default_storage.save va faire le taf tout seul, et retourner le nom de fichier qu'il aura utilisé).

Je laisse Benjamin se prononcer sur create_first_name_last_name_attributes → create_custom_attributes, de mon côté pas sûr qu'on veuille taper une image dans le profil par défaut.

Je ne suis pas sûr que toutes ces modifications pour passer request soient utiles. (j'aimerais penser que non)

#32 - 19 septembre 2018 15:01 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Frédéric Péters a écrit :

> (c'est vraiment juste mon confort mais tu peux taper des images plus petites pour les tests ? (genre dans gimp dessine un carré bleu et un carré rouge et hop))

Oui bien sûr.

> Je pensais moi vraiment à un {% thumbnail attribute.values.0 as thumb_im %}, pas de construction temporaire inutile d'URL.

Ok, avec l'utilisation de get_thumbnail dans la définition de la balise de gabarit, dans cette nouvelle version du patch, on arrive à un {% thumbnail attribute.values.0 as thumb_im %}.

> (il faut des guillemets autour de la classe,

Aïe. Merci.

> et curieux tu as un espace insécable derrière le premier {{. (je te suggérerais de configurer ton éditeur pour les rendre visibles)

Double aïe. Re-merci. (Je croyais que nbsp était dans les listchars de ma conf vim, c'est corrigé).

> Je reste sur mon avis que _store_image peut être réduit, davantage encore. (et je dirais aussi qu'on se fout du hash de l'image)

(default_storage.save va faire le taf tout seul, et retourner le nom de fichier qu'il aura utilisé).

Ok, j'ai réduit encore. En adaptant _delete_image_from_user de façon similaire.

Je laisse Benjamin se prononcer sur create_first_name_last_name_attributes → create_custom_attributes, de mon côté pas sûr qu'on veuille taper une image dans le profil par défaut.

J'ai rétabli le code initial (et donc pas d'image d'avatar dans le profil par défaut).

Je ne suis pas sûr que toutes ces modifications pour passer request soient utiles. (j'aimerais penser que non)

En effet, les middleware incluent in StoreRequestMiddleware, qui rend toutes ces modifications superflues.

#33 - 19 septembre 2018 15:19 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Encore une correction impromptue, pardon :
la sérialisation doit renvoyer un chemin local, et la désérialisation doit prendre en entrée ce chemin et renvoyer l'URL publique correspondant.

Je pose le patch corrigé ici.

Edit: l'interdiff de la correction est

```
diff --git a/src/authentic2/utils.py b/src/authentic2/utils.py
index 2d2c5c00..2c712a3e 100644
--- a/src/authentic2/utils.py
+++ b/src/authentic2/utils.py
@@ -1118,21 +1118,21 @@ def _delete_images_from_user(owner_pk, attr_label):

 def image_serialize(image, owner_uuid, owner_pk, attr_label):
-    uri = ''
+    img_media_path = ''
     if isinstance(image, basestring):
-        uri = image
+        img_media_path = image
     else:
         # Discard previous user avatars
         _delete_images_from_user(owner_pk, attr_label)
         if image:
             img_media_path = _store_image(image, owner_uuid, attr_label)
-            uri = os.path.join(settings.MEDIA_URL, img_media_path)
-    return uri
+    return img_media_path

-def image_deserialize(image_uri):
+def image_deserialize(img_media_path):
     from authentic2.middleware import StoreRequestMiddleware

-    if image_uri:
+    if img_media_path:
         request = StoreRequestMiddleware().get_request()
-        return request.build_absolute_uri(image_uri)
+        img_uri = os.path.join(settings.MEDIA_URL, img_media_path)
+        return request.build_absolute_uri(img_uri)
```

#34 - 19 septembre 2018 15:27 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Paul Marillonnet a écrit :

la sérialisation doit renvoyer un chemin local, et la désérialisation doit prendre en entrée ce chemin et renvoyer l'URL publique correspondant.

Ce qui permet de retirer les deux lignes bien disgracieuses (et dangereuses dans l'ancienne version du patch, car ça casse la désérialisation si jamais le MEDIA_URL change) :

```
diff --git a/src/authentic2/utils.py b/src/authentic2/utils.py
```

```
index 2c712a3e..923e4c1a 100644
--- a/src/authentic2/utils.py
+++ b/src/authentic2/utils.py
@@ -1111,9 +1111,7 @@ def _delete_images_from_user(owner_pk, attr_label):

     for value in values:
         if value.content:
-            # Direct URI <-> file location correspondence
-            local_file = value.content.split(default_storage.base_url)[-1]
-            default_storage.delete(local_file)
+            default_storage.delete(value.content)
         value.delete()
```

Edit:
il y a quand même un split à introduire, lorsque l'usager valide le formulaire dans lequel il conserve son image (le sérialisateur reçoit alors l'URL publique de l'image, et il doit retrouver le chemin relatif au default_storage à partir de cette URL).
je finis de valider les tests et je soumets la version à jour du patch.

#35 - 19 septembre 2018 18:43 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Paul Marillonnet a écrit :

> je finis de valider les tests et je soumets la version à jour du patch.

Voilà, avec en prime un urllib.unquote pour restituer les caractères échappés dans l'URL (accents et espaces notamment).
Et un changement de nom de fichier d'images de tests pour vérifier que le unquote fonctionne.

#36 - 26 septembre 2018 13:19 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Paul Marillonnet a écrit :
>
> > je finis de valider les tests et je soumets la version à jour du patch.
>
> Voilà, avec en prime un urllib.unquote pour restituer les caractères échappés dans l'URL (accents et espaces notamment).
> Et un changement de nom de fichier d'images de tests pour vérifier que le unquote fonctionne.

- il y a déjà une gestion des valeurs par défaut dans app_settings.py, ceci me parait inutile:

```
DEFAULT_IMAGE_DIMENSIONS = "150x150"
DEFAULT_IMAGE_CROPPING = "center"
DEFAULT_IMAGE_QUALITY = 99
...
def get_image_thumbnail_parameters():
    return (app_settings.A2_ATTRIBUTE_KIND_IMAGE_DIMENSIONS or DEFAULT_IMAGE_DIMENSIONS,
        app_settings.A2_ATTRIBUTE_KIND_IMAGE_CROPPING or DEFAULT_IMAGE_CROPPING,
        app_settings.A2_ATTRIBUTE_KIND_IMAGE_QUALITY or DEFAULT_IMAGE_QUALITY)
```

- [redacted]

```
    template_with_initial = (
        '%(initial_text)s: <br /> <img src="%(initial)s"/> <br />'
        '%(clear_template)s<br />%(input_text)s: %(input)s'
    )
```

  tu peux mettre un if django.VERSION < (1, 9): comme ça on est sûr que ça ne sera plus pris en compte ensuite, soit autour de cette seule déclaration soit autour de deux déclarations différentes de la même classe (le reste me paraissant Django 1.1x only).

- [redacted]

```
        url(r'^media/(?P<path>.*)$', media_serve, {
        'document_root': settings.MEDIA_ROOT})
    ]
```

  pas possible trop lâche, on ne souhaite pas servir tout le contenu de media pour toujours

- je ne pense pas que passer des paramètres à store_image soit nécessaire on a une image, on génère un chemin, on la sauve, on stocke le chemin dans l'attribut, le nettoyage se fait simplement en listant les images, listant les chemins stockés dans des attributs du type image, prendre la différence, supprimer

Première relecture mais je vais repasser dessus dans l'aprem plus attentivement.

#37 - 28 septembre 2018 17:19 - Paul Marillonnet

Benjamin Dauvergne a écrit :

- [...] pas possible trop lâche, on ne souhaite pas servir tout le contenu de media pour toujours

Je me suis basé sur ce qui était déjà en place pour servir les fichiers statiques (en mode DEBUG seulement).
Pourquoi est-ce trop lâche ?

- je ne pense pas que passer des paramètres à store_image soit nécessaire on a une image, on génère un chemin, on la sauve, on stocke le chemin dans l'attribut,

Ok. Je ne vois pas encore comment la construction du chemin peut se faire sans passage de paramètres (on a besoin de l'uuid de l'usager, ainsi que du libellé d'attribut).

Ok pour le reste. je pose un nouveau patch dès que j'ai compris comment faire pour ces deux points me posant encore problème.

#38 - 28 septembre 2018 18:09 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Benjamin Dauvergne a écrit :
>
> - [...] pas possible trop lâche, on ne souhaite pas servir tout le contenu de media pour toujours
>
> Je me suis basé sur ce qui était déjà en place pour servir les fichiers statiques (en mode DEBUG seulement).
> Pourquoi est-ce trop lâche ?

Ok désolé pas vu que c'était uniquement pour du debug, tu peux oublier.

- je ne pense pas que passer des paramètres à store_image soit nécessaire on a une image, on génère un chemin, on la sauve, on stocke le chemin dans l'attribut,

> Ok. Je ne vois pas encore comment la construction du chemin peut se faire sans passage de paramètres (on a besoin de l'uuid de l'usager, ainsi que du libellé d'attribut).

Pourquoi ? Le chemin ça devrait être /images/<hash-image>.<extension-si-possible> il n'est pas nécessaire de lier ça à l'attribut ou à l'uuid de l'usage, Django s'occupe de toute façon d'éviter les collisions.

Il faut vérifier que sur suppression d'un compte et de ses attributs, l'image est supprimé, idem sur modification de l'image, l'image d'avant doit partir.

#39 - 01 octobre 2018 09:23 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Penser à stocker les fichiers dans avatars/<uuid>[0:4]/<uuid>.jpeg pour éviter d'avoir 80 000 fichiers dans le même répertoire.

On abandonne ça alors ?

> Il faut vérifier que sur suppression d'un compte et de ses attributs, l'image est supprimé, idem sur modification de l'image, l'image d'avant doit partir.

Oui, ok, je comprends.

Mais le problème, je crois, est le suivant :
- utilisatrice A charge `sardou.png` comme image de profil. Le fichier est stocké dans avatars/7dhziu3678kl83746hdu.png
- utilisateur B charge la même image. Même valeur de hachage donc, et même fichier sur serveur.
- utilisateur B supprime son image d'avatar. Utilisatrice A perd aussi son image de profil.

Deux possibilités donc, je crois, pour remédier à ça :
- rester sur l'ancien format chemin de fichier, pour lequel l'uuid de l'utilisateur apparaît. (Avec aussi le label de l'attribut apparaissant dans le chemin, parce qu'un utilisateur peut décider de charger plusieurs fois la même image pour plusieurs attributs différents. Et la suppression de l'une des images ne doit pas entraîner la suppression des autres par effet de bord).
- ajouter un champ compteur used_count pour recenser le nombre d'utilisateurs possédant cet AttributeValue. L'AttributeValue d'image et le contenu associé ne sont supprimés que lorsque le compteur vaut 0.

Je resterais bien sur la première option, ça me paraît plus simple, parce que le cas des doublons d'image va être rare je pense (à moins que l'on décide de mettre en place une image d'avatar par défaut). Je me plante ?

Mais sinon, oui, dans tous les cas, la suppression d'un compte ou d'un attribut doit être prise en compte pour la gestion des images.
Je vais modifier le code, et rajouter les tests qui vont avec. Merci, bien vu :)

#40 - 01 octobre 2018 10:01 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Benjamin Dauvergne a écrit :
>
> > Penser à stocker les fichiers dans avatars/<uuid>[0:4]/<uuid>.jpeg pour éviter d'avoir 80 000 fichiers dans le même répertoire.
>
> On abandonne ça alors ?

Non mais on s'est mal compris alors, je n'ai jamais dit d'utiliser l'uuid mais le hash du fichier donc

```
h = sha1(image.read()).hexdigest()
path = '%s/%s.%s' % (h[:3], h[3:], extension)
```

> > Il faut vérifier que sur suppression d'un compte et de ses attributs, l'image est supprimé, idem sur modification de l'image, l'image d'avant doit partir.
>
> Oui, ok, je comprends.
>
> Mais le problème, je crois, est le suivant :
> - utilisatrice A charge `sardou.png` comme image de profil. Le fichier est stocké dans avatars/7dhziu3678kl83746hdu.png
> - utilisateur B charge la même image. Même valeur de hachage donc, et même fichier sur serveur.

Ne peut pas arriver, django.core.storage renomme les fichiers en cas de collision, lis en la doc.

> Mais sinon, oui, dans tous les cas, la suppression d'un compte ou d'un attribut doit être prise en compte pour la gestion des images.
> Je vais modifier le code, et rajouter les tests qui vont avec. Merci, bien vu :)

J'ai lu un peu stackoverflow&co sur ce sujet, les champs fichiers ne sont jamais nettoyé sur suppression du modèle ou mise à jour parce que ça ne marche pas avec les transactions:

```
* begin
* suppression du modèle
* suppression du modèle
* rollback (on se retrouve avec un champ fichier qui n'existe plus)
```

L'application django-cleanup règle le problème mais uniquement à partir de Django 1.9 via nouveau callback "on_commit" qui est quand une transaction est committée, je pense qu'ils proposent aussi une commande de management qui nettoie les fichiers. À voir si on peut intégrer cette dépendance.

#41 - 01 octobre 2018 10:08 - Frédéric Péters

> L'application django-cleanup règle le problème mais uniquement à partir de Django 1.9 via nouveau callback "on_commit" qui est quand une transaction est committée, je pense qu'ils proposent aussi une commande de management qui nettoie les fichiers. À voir si on peut intégrer cette dépendance.

On utilise encore Django 1.8.

#42 - 01 octobre 2018 10:13 - Benjamin Dauvergne

django-cleanup fonctionne en Django 1.8 mais avec un bug en cas de transaction rollbacké (on_commit = lambda x: x()).

#43 - 01 octobre 2018 10:19 - Benjamin Dauvergne

Bon je dis n'importe quoi on a pas de champ fichier ici ça ne nous aidera pas, oublie les problèmes de suppression de fichiers, on reviendra dessus plus tard, par contre ce serait bien de ne pas appeler le répertoire avatar, on aura d'autres types d'images peut-être un jour.

#44 - 01 octobre 2018 10:25 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Ne peut pas arriver, django.core.storage renomme les fichiers en cas de collision, lis en la doc.

Yes, j'avais oublié ça, merci.

> oublie les problèmes de suppression de fichiers, on reviendra dessus plus tard, par contre ce serait bien de ne pas appeler le répertoire avatar, on aura d'autres types d'images peut-être un jour.

Ok.
Le dossier parent porte le nom du libellé d'attribut. Je vas changer ça puisque, comme tu me le fais remarquer, il n'y aura pas de collision avec le default_storage sur le noms des images stockées.
(Il restait un commentaire dans utils.image_serialize qui laissait croire que le code est spécialement prévu pour cet attribut d'image d'avatar, c'est corrigé).

#45 - 01 octobre 2018 17:28 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> le nettoyage se fait simplement en listant les images, listant les chemins stockés dans des attributs du type image, prendre la différence, supprimer

Est-ce qu'on souhaite encore supprimer les images précédemment stockées lorsque l'utilisateur charge une nouvelle image ou bien supprime l'image courante ?
Deux options :

- Si c'est toujours le cas : on doit supprimer toutes les images pour un attribut donné seulement. Et donc la fonction de suppression prend toujours le libellé d'attribut en paramètre (cf patch joint, authentic2.utils._delete_images_from_user)
- Si on oublie complètement cette problématique de suppression : on vire ce paramètre, ainsi que le dico serialize_eval_kwargs, ce qui ne sera pas plus mal pour la clarté du code.

#46 - 01 octobre 2018 17:29 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Et donc, le patch, pardon.

#47 - 03 octobre 2018 10:44 - Paul Marillonnet

Paul Marillonnet a écrit :

> - Si on oublie complètement cette problématique de suppression : on vire ce paramètre, ainsi que le dico serialize_eval_kwargs, ce qui ne sera pas plus mal pour la clarté du code.

Je dis des bêtises : cette option n'est pas envisageable en l'état, parce que c'est en maintenant une unique valeur d'attribut pour un attribut image donné que l'on retrouve, à la génération des vues /accounts/*, l'image courante pour cet attribut.

#48 - 04 octobre 2018 11:05 - Paul Marillonnet

J'ai pu découvrir la capacité de Benj à relire des patchs à voix haute.
En gros, on va simplifier :
- le type d'attribut (attribut_kind) va être directement 'avatar', et non pas image. On n'effectue plus de distinction les différents Attribute de ce même type.
- on oublie dans ce patch toute problématique de suppression. (Pour un attribut non-multiple, l'unicité sur <utilisateur ; valeur d'attribut> permet de dégager cette problématique de suppression, au moins pour l'instant.)

Donc, les modifications n'impliquent quasiment que des lignes à supprimer. Je m'en occupe maintenant et re-soumets ici un patch aujourd'hui.

#49 - 04 octobre 2018 11:57 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Allez hop.

#50 - 04 octobre 2018 12:50 - Benjamin Dauvergne

- Je ne comprends pas les changements dans test_manager et test_profile ? D'où viennent-ils ?
- Plutôt une fixture media comme cela:

```
@pytest.fixture
```

```
def media(settings, tmpdir):
    settings.MEDIA_ROOT = str(tmpdir.mkdir('media'))
```

- C'est quoi le besoin de monkeyrequest ?
- Je vois que tu te prends le choux pour faire une URI absolu au niveau de deserializer (via StoreRequestMiddleware.get_request()), ne le fais pas, génère du relatif, on produira de l'absolu quand ça sort (ou pas, charge au client de se débrouiller)

Et je m'arrête pour continuer plus tard.

#51 - 04 octobre 2018 15:11 - Paul Marillonnet

Paul Marillonnet a écrit :

> - vérifier encore qu'il y a bien un bogue dans webtest (version 2.0.30) lors de l'annuliation (`form.submit(cancel)`) d'un formulaire encodé en multipart/form-data. Le code de webtest.forms.submit_fields semble (tout comme celui de webtest.app.encode_multipart) attendre une valeur de champ non nulle pour retrouver ce champ avant encodage du formulaire. Mais il teste l'égalité de la valeur contre webtest.forms.Submit.value_if_submitted() qui vaut toujours None.
> Cf en particulier ce code de webtest.forms :
> [...]

Je suis resté là dessus, pas moyen de faire tourner les tests avec un formulaire envoyé en "multipart/form-data"

#52 - 04 octobre 2018 15:32 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> - C'est quoi le besoin de monkeyrequest ?

Je comprends maintenant ce qui ne va pas : le désérialisateur de valeur d'attribut avait besoin d'accéder à un objet de requête (WSGIRequest). Ça rejoint ta remarque suivante : ne pas se prendre la tête à manipuler des URIs absolues dans la désérialisation.

#53 - 04 octobre 2018 15:33 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Paul Marillonnet a écrit :
>
> > - vérifier encore qu'il y a bien un bogue dans webtest (version 2.0.30) lors de l'annuliation (`form.submit(cancel)`) d'un formulaire encodé en multipart/form-data. Le code de webtest.forms.submit_fields semble (tout comme celui de webtest.app.encode_multipart) attendre une valeur de champ non nulle pour retrouver ce champ avant encodage du formulaire. Mais il teste l'égalité de la valeur contre webtest.forms.Submit.value_if_submitted() qui vaut toujours None.

Pas du tout, il le retrouve via l'index qui par défaut vaut 0:

```
    def submit_fields(self, name=None, index=None, submit_value=None):
...
        # If no particular button was selected, use the first one
        if index is None and submit_value is None:
            index = 0
...
            if submit_name is not None and name == submit_name:
                if index is not None and current_index == index:
                    submit.append((field.pos, name, field.value_if_submitted()))
```

Ça doit marcher quelque soit l'encodage, tu as tracé dans ce code pour être sûr d'avoir trouvé un bug ? De plus je ne vois pas de problème de ce genre signalé sur le projet webtest, c'est tellement gros que je pense que quelqu'un serait tombé dessus quand même, su un projet aussi utilisé.

#54 - 04 octobre 2018 16:19 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Ça doit marcher quelque soit l'encodage, tu as tracé dans ce code pour être sûr d'avoir trouvé un bug ? De plus je ne vois pas de problème de ce genre signalé sur le projet webtest, c'est tellement gros que je pense que quelqu'un serait tombé dessus quand même, su un projet aussi utilisé.

Je n'arrive plus à reproduire. Je crois que l'inclusion d'un attribut avatar par défaut, dans mes précédents patchs, faisait échouer cette partie des tests, à savoir le remplissage du formulaire de gestion du compte.

#55 - 04 octobre 2018 16:35 - Benjamin Dauvergne

J'ai reproduit, j'ai ouvert un ticket chez webtest et je vais leur soumettre une pull-request, en attendant faut bidouiller en mettant " dans form.fields['cancel']._value@.

https://github.com/Pylons/webtest/issues/205

#56 - 04 octobre 2018 17:21 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Benjamin Dauvergne a écrit :
>
> > Ça doit marcher quelque soit l'encodage, tu as tracé dans ce code pour être sûr d'avoir trouvé un bug ? De plus je ne vois pas de problème de ce genre signalé sur le projet webtest, c'est tellement gros que je pense que quelqu'un serait tombé dessus quand même, su un projet aussi utilisé.
>
> Je n'arrive plus à reproduire. Je crois que l'inclusion d'un attribut avatar par défaut, dans mes précédents patchs, faisait échouer cette partie des tests, à savoir le remplissage du formulaire de gestion du compte.

Oui c'était ça.

#57 - 04 octobre 2018 18:12 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Ok, j'inclus ta bidouille dans les tests (qui, je crois, servira si on décide de rajouter un champ avatar par défaut dans le profil utilisateur).

Et donc je nomme le bouton submit, dans le template de la page de réinitialisation du mot de passe, pour pouvoir le retrouver dans tests/test_manager.py::test_manager_user_password_reset

(avec une explication dans le message de commit, parce qu'on est loin de la fonctionnalité demandée dans le ticket, quand même)

#58 - 04 octobre 2018 18:49 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Ok, j'inclus ta bidouille dans les tests (qui, je crois, servira si on décide de rajouter un champ avatar par défaut dans le profil utilisateur).
>
> Et donc je nomme le bouton submit, dans le template de la page de réinitialisation du mot de passe, pour pouvoir le retrouver dans tests/test_manager.py::test_manager_user_password_reset
>
> (avec une explication dans le message de commit, parce qu'on est loin de la fonctionnalité demandée dans le ticket, quand même)

Encore une incompréhension de ma part, la bidouille je la pensais nécessaire pour tes nouveaux tests, non il n'en faut pas dans les tests qui passent déjà tout seul.

J'ai encore un peu de mal avec le traitement explicite du champ dans le rendu du profil, il faut trouver mieux, je vais voir.

#59 - 04 octobre 2018 19:00 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Encore une incompréhension de ma part, la bidouille je la pensais nécessaire pour tes nouveaux tests, non il n'en faut pas dans les tests qui passent déjà tout seul.

Ok, pas de souci, incompréhension de ma part aussi. Je corrige ça.

> J'ai encore un peu de mal avec le traitement explicite du champ dans le rendu du profil, il faut trouver mieux, je vais voir.

Oui, je comprends ce que tu veux dire. Je vais chercher de mon côté.

#60 - 11 octobre 2018 09:42 - Paul Marillonnet

> J'ai encore un peu de mal avec le traitement explicite du champ dans le rendu du profil, il faut trouver mieux, je vais voir.

À vue de nez, j'aimerais bien déplacer une partie du gabarit src/authentic2/templates/authentic2/accounts.html dans un fichier à part accounts_form_field.html, et utiliser la balise {% include '...' %}.
Ça simplifierait le fichier accounts.hmtl qui est déjà assez long, et c'est le plus simple que je vois comme ça (càd conserver le traitement explicite du

champ, mais le mettre dans un fichier à part).

#61 - 11 octobre 2018 09:57 - Paul Marillonnet

Juste une question de lisibilité, mais est-ce déjà un peu mieux comme ça ?

#62 - 11 octobre 2018 09:58 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Avec le patch donc…

#63 - 11 octobre 2018 14:09 - Frédéric Péters

(rebase, il ne s'applique pas sur master, dans tests/test_attribute_kinds.py) (ou pointe vers une branche)

#64 - 11 octobre 2018 14:25 - Frédéric Péters

Après avoir mis une photo,

```
[2018-10-11 Thu 14:10:57] - - - ERROR hobo.agent.authentic2.provisionning.do_provision: error in provisionning
 thread
Traceback (most recent call last):
  File "/home/fred/src/eo/hobo/hobo/agent/authentic2/provisionning.py", line 254, in do_provision
    self.notify_users(ous, saved.get(self.User, []))
  File "/home/fred/src/eo/hobo/hobo/agent/authentic2/provisionning.py", line 161, in notify_users
    'data': [user_to_json(None, user, user_roles) for user in users],
  File "/home/fred/src/eo/hobo/hobo/agent/authentic2/provisionning.py", line 93, in user_to_json
    data.update(BaseUserSerializer(user).data)
  File "/home/fred/src/eo/venv1.11/local/lib/python2.7/site-packages/rest_framework/serializers.py", line 534,
 in data
    ret = super(Serializer, self).data
  File "/home/fred/src/eo/venv1.11/local/lib/python2.7/site-packages/rest_framework/serializers.py", line 263,
 in data
    self._data = self.to_representation(self.instance)
  File "/home/fred/src/eo/venv1.11/local/lib/python2.7/site-packages/rest_framework/serializers.py", line 501,
 in to_representation
    ret[field.field_name] = field.to_representation(attribute)
  File "/home/fred/src/eo/authentic/src/authentic2/attribute_kinds.py", line 116, in to_representation
    return self.context['request'].build_absolute_uri(value)
KeyError: 'request'
```

Je pensais qu'après "Ça rejoint ta remarque suivante : ne pas se prendre la tête à manipuler des URIs absolues dans la désérialisation." il n'y aurait plus ce genre de truc.

Le chemin vers l'image (que je vois dans /manage/users/123/) est :

```
/media/photo/dd81//dd816425f7384718b784d08127f219e5bace723223ce8a31e13a18cfe37e3b42.PNG
```

Deux //, et PNG en majuscules.

J'imaginais sorl.thumbnail utilisé pour stocker une miniature, ou au moins pour la servir, en l'état ce n'est pas le cas, résultat il suffirait pour un usager d'uploader un fichier délictueux et d'un bug pour exploiter la connexion d'un admin visitant le /manage/.

Je reste fasciné par _store_image. (je commence aussi à me dire que tu n'as pas attaché le dernier patch ?).

```
/media/photo/dd81//dd816425f7384718b784d08127f219e5bace723223ce8a31e13a18cfe37e3b42.PNG
```

c'est aussi ce qui se trouve repris dans l'attribut SAML associé, il faudrait là avoir l'URL absolue (je trouve).

#65 - 11 octobre 2018 17:46 - Paul Marillonnet

Frédéric Péters a écrit :

> (je commence aussi à me dire que tu n'as pas attaché le dernier patch ?).

Je n'arrive pas à reproduire, et les lignes de la stack trace ne correspondent pas au code du dernier patch.

Par exemple

```
  File "/home/fred/src/eo/authentic/src/authentic2/attribute_kinds.py", line 116, in to_representation
```

n'est pas raccord avec le patch.

Ça concorde en revanche avec des anciennes versions du patch (celle du 17 sept., par exemple).

#66 - 11 octobre 2018 17:55 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Frédéric, tes remarques qui, je crois, sont toujours valables pour la dernière version du patch, et que je corrige dès que possible :
- l'extension du fichier à passer en lettres minuscules
- l'URL absolue dans l'attribut SAML

Pour le thumbnail, et le double slash, il me semble que c'est bon dans la dernière version, mais je vais vérifier ça aussi.

Je viens aussi de voir que quelques glitches sont présents en Django 1.8 seulement, je vais corriger aussi.

#67 - 11 octobre 2018 17:59 - Frédéric Péters

Ok j'ai du m'emmêler sur le patch vu qu'il ne s'appliquait pas et piocher ensuite une ancienne copie :/

#68 - 11 octobre 2018 18:03 - Frédéric Péters

> Pour le thumbnail, et le double slash, il me semble que c'est bon dans la dernière version, mais je vais vérifier ça aussi.

Pour le double slash c'est ok. Mais c'est toujours le fichier uploadé qui est servi, pas une miniature.

#69 - 12 octobre 2018 09:32 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> - l'URL absolue dans l'attribut SAML

Je m'en occuperai après, continuons sur les aspects front ici.

#70 - 12 octobre 2018 11:54 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

- Statut changé de En cours à Solution proposée

Frédéric Péters a écrit :

> Ok j'ai du m'emmêler sur le patch vu qu'il ne s'appliquait pas et piocher ensuite une ancienne copie :/

Ma faute, désolé, j'ai oublié de rebaser.

> Mais c'est toujours le fichier uploadé qui est servi, pas une miniature.

Corrigé.
Corrigé aussi la casse de l'extension du fichier d'image, et les glitches en Django 1.8.

La distinction entre django 1.8 et versions supérieures, dans le code du widget, se fait aussi sur la définition d'une méthode, en plus de l'attribut tel que dans les patches précédents. Je trouve le code plus lisible comme ça, et ça faciliterait sans doute le débogage, mais je me plante peut-être (cf authentic2.forms.widgets.ProfileImageInput).

Benjamin Dauvergne a écrit :

> Je m'en occuperai après, continuons sur les aspects front ici.

Ok.

#71 - 22 octobre 2018 15:09 - Benjamin Dauvergne

- Fichier 0001-add-an-html_value-method-to-attribut-kinds.patch ajouté

Je n'aime pas la customisation du template accounts.html pour le rendu des images de profil, je préfère avoir une méthode au niveau du kind.

#72 - 22 octobre 2018 15:11 - Benjamin Dauvergne

Aussi l'import django pour accéder à django.VERSION devrait remonter en haut de fichier, tu peux intégrer mes modifications ?

#73 - 22 octobre 2018 15:33 - Paul Marillonnet

Merci. Je suis en train de faire tourner les tests unitaires, et je fais quelques tests d'utilisation dans mon environnement Publik local. Une fois que c'est bon, je pose le patch incluant tes modifications.

Je me suis permis un petite modif du code que tu fournis, sinon ça plante :

```
diff --git a/src/authentic2/views.py b/src/authentic2/views.py
index d39419b4..e9e296f5 100644
--- a/src/authentic2/views.py
+++ b/src/authentic2/views.py
@@ -444,7 +444,7 @@ class ProfileView(cbv.TemplateNamesMixin, TemplateView):
             if attribute:
                 if not attribute.user_visible:
                     continue
-                html_value = attribute.get_kind().get('html_value', lambda a, b: b)
+                html_value = attribute.get_kind().get('html_value', lambda x: x)
                 qs = models.AttributeValue.objects.with_owner(request.user)
                 qs = qs.filter(attribute=attribute)
                 qs = qs.select_related()
```

#74 - 22 octobre 2018 15:44 - Paul Marillonnet

Edit: mauvaise modif de ma part : la modif à faire n'est pas dans la définition de la fonction lambda, mais dans l'appel à map, plus loin dans le coe. Je corrige.

#75 - 22 octobre 2018 16:23 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

J'ai remanié un peu le code du patch que tu proposes :

- il n'y a plus d'appel à map, simplement une listcomp
- profile_image_html_value fait appel à get_image_thumbnail, et prend donc la requête en cours en paramètre.
- la fonction lambda pour les types d'attribut ne définissant pas de html_value prend donc trois paramètres.
  J'ai tout inclus dans un seul patch.

#76 - 22 octobre 2018 16:33 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> J'ai remanié un peu le code du patch que tu proposes :
>
> - il n'y a plus d'appel à map, simplement une listcomp

Ok.

> - profile_image_html_value fait appel à get_image_thumbnail, et prend donc la requête en cours en paramètre.

Si ça marchait sans requête, c'est que celle-ci est inutile, je ne vois pas bien à quoi sert get_image_thumbnail (ça ressemble à la même fonction de sorl-thumbnail avec request en plus), pourquoi veux-tu une URL absolue ici ?

#77 - 22 octobre 2018 17:02 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Si ça marchait sans requête, c'est que celle-ci est inutile, je ne vois pas bien à quoi sert get_image_thumbnail (ça ressemble à la même fonction de sorl-thumbnail avec request en plus), pourquoi veux-tu une URL absolue ici ?

Ce n'est pas la requête en elle-même qui me semble nécessaire, mais plutôt fournir l'URL absolue du fichier d'image d'origine.
Dans mes derniers tests, fournir un chemin relatif et non l'URL absolue du fichier d'origine aboutissait à la création d'un objet incomplet, n'ayant pas d'attribut url.
(et donc il devient impossible de faire quelque chose thumb.url dans le code.)

Je suis en train de reproduire, je reviens vers toi avec la confirmation de ce comportement, ou bien une correction du patch (dans le code où je me trompe ici).

#78 - 22 octobre 2018 17:50 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Je me suis trompé, j'ai dû louper quelque chose lors de l'écriture de cette partie du code.
Une version corrigée, comme indiqué plus un peu tôt ici.

Autre chose encore : pour l'instant, on n'utilise le thumbnail que pour les pages html servies, par pour les API /user/ et /users/.
Est-ce que c'est le comportement que l'on souhaite ? Ou bien est-ce qu'on sert le thumbnail partout ?

#79 - 22 octobre 2018 17:55 - Frédéric Péters

> Est-ce que c'est le comportement que l'on souhaite ? Ou bien est-ce qu'on sert le thumbnail partout ?

On doit servir une image saine partout. (donc oui, thumbnail).

#80 - 22 octobre 2018 18:23 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Frédéric Péters a écrit :

> On doit servir une image saine partout. (donc oui, thumbnail).

Une première version où l'on sert le thumbnail partout.

Je ne sais pas si on a intérêt à stocker directement le thumbnail. Je me mets tout de suite à écrire une version alternative du patch, pour lequel on stocke directement et seulement le thumbnail.

#81 - 22 octobre 2018 18:28 - Frédéric Péters

> Je ne sais pas si on a intérêt à stocker directement le thumbnail. Je me mets tout de suite à écrire une version alternative du patch, pour lequel on stocke directement et seulement le thumbnail.

À mon sens c'est utile de conserver l'image d'origine, si jamais on décide de modifier la taille de la vignette, genre.

#82 - 22 octobre 2018 19:26 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Frédéric Péters a écrit :

> > Je ne sais pas si on a intérêt à stocker directement le thumbnail. Je me mets tout de suite à écrire une version alternative du patch, pour lequel on stocke directement et seulement le thumbnail.
>
> À mon sens c'est utile de conserver l'image d'origine, si jamais on décide de modifier la taille de la vignette, genre.

Ok, donc la version à jour du code.
Toutes les images servies sont des thumbnail.
Et j'ai altéré la contrainte de non-nullité de AttributeValue.content, car je trouve ça plus organique ainsi (par exemple, quand l'usager supprime son image de profil, alors le contenu de le valeur d'attribut associée vaut None et non pas la chaîne vide).

#83 - 22 octobre 2018 19:29 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Retiré l'import de authentic2.utils dans authentic2.forms.widgets, car il en sert plus à rien.

Edit: Re-testé cette nuit, j'avais encore de problème de slash multiples dans les URL. Je regarde ça maintenant.

Edit2: Il s'agit en fait d'une 404 lorsque l'usager clique sur l'image.

Edit3:
Paul Marillonnet a écrit :

> Et j'ai altéré la contrainte de non-nullité de AttributeValue.content, car je trouve ça plus organique ainsi (par exemple, quand l'usager supprime

> son image de profil, alors le contenu de le valeur d'attribut associée vaut None et non pas la chaîne vide).

Et donc j'ai des changements unstaged pour les tests unitaires, oubliés dans mon dernier patch.. Je corrige ça maintenant.

#84 - 23 octobre 2018 09:36 - Paul Marillonnet

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Et donc le patch à jour (avec les trois corrections mentionnées dans les trois "Edit" de mon dernier commentaire ici).

#85 - 23 octobre 2018 11:31 - Benjamin Dauvergne

Je relis.

#86 - 23 octobre 2018 11:42 - Benjamin Dauvergne

Bon j'ai du mal à suivre car l'objet à changer avec les interventions de Fred, donc maintenant en fait l'avatar ce n'est pas l'image uploadé par l'utilisateur dont on se fout mais uniquement la thumbnail générée ? Dans ce cas il faudrait utiliser sorl-thumbnail directement à l'upload et ne pas conserver l'image uploadé qui ne sert à rien. Je vais voir pour faire cela.

#87 - 23 octobre 2018 11:55 - Frédéric Péters

Paul pose cette question et je lui réponds que conserver l'image d'origine a un sens (commentaire 81).

Mais peu importe la forme, je ne commente plus, on a juste besoin de ce ticket, et de la distribution de l'info dans les attributs, aujourd'hui en recette.

#88 - 23 octobre 2018 12:52 - Benjamin Dauvergne

Ce ne sera pas aujourd'hui en recette, j'ai commencé une réécriture sans sorl-thumbnail:

http://git.entrouvert.org/authentic.git/log/?h=wip/26022-photo-de-profil-avatar-dans-le-profil

#89 - 23 octobre 2018 13:11 - Frédéric Péters

> Ce ne sera pas aujourd'hui en recette, j'ai commencé une réécriture sans sorl-thumbnail:

J'ai noté que je n'intervenais plus mais sérieusement, ça fait des semaines que c'est annoncé, reporté, et alors qu'une nouvelle échéance a été donnée pour une mise en prod dans deux jours il n'y a même plus de patch **du tout** ?

#90 - 23 octobre 2018 13:46 - Benjamin Dauvergne

Frédéric Péters a écrit :

> > Ce ne sera pas aujourd'hui en recette, j'ai commencé une réécriture sans sorl-thumbnail:
>
> J'ai noté que je n'intervenais plus mais sérieusement, ça fait des semaines que c'est annoncé, reporté, et alors qu'une nouvelle échéance a été donnée pour une mise en prod dans deux jours il n'y a même plus de patch **du tout** ?

Le code n'est pas acceptable en l'état donc non il n'y a plus de patch, j'étais ok pour stocker des fichiers tels quels là c'est devenu n'importe quoi. Concernant une fonctionnalité totalement inutile, je ne laissera pas passer.

#91 - 23 octobre 2018 14:30 - Paul Marillonnet

Je... ne sais pas quoi dire.

Est-ce qu'il y a encore quelque chose que je puisse faire pour aider ? (mis à part y mettre du mien pour monter en niveau sur le dév Python, puisque de toute évidence ce n'est pas encore ça)

#92 - 23 octobre 2018 14:52 - Benjamin Dauvergne

- Assigné à changé de Paul Marillonnet à Benjamin Dauvergne

Paul Marillonnet a écrit :

> Je... ne sais pas quoi dire.
>
> Est-ce qu'il y a encore quelque chose que je puisse faire pour aider ? (mis à part y mettre du mien pour monter en niveau sur le dév Python, puisque de toute évidence ce n'est pas encore ça)

Ce n'est pas ta faute, c'est juste que tout le monde met son grain de sel et ça part en sucette.

#93 - 23 octobre 2018 17:38 - Benjamin Dauvergne

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Voila c'est poussable et il y a le support dans l'API, c'était finalement rien à ajouter grâce au support d'URL absolues dans le champ fichier de DRF.

J'ai deux questions :

- j'ai gardé tes deux tests Paul, mais il me semble faire la même chose non ? Ne pourrait-on garder que le test avec deux champs ?
- j'ai supprimé l'utilisation de sorl-thumbnail, je fais directement un image.crop((0, 0, 150, 150)) via pillow, ça m'irait de ne rien faire et de simplement refuser les images plus grandes que 150x150 (et je remarque une API Image.thumbnail() qui n'a pas l'air mal non plus).

#94 - 23 octobre 2018 17:56 - Frédéric Péters

> j'ai supprimé l'utilisation de sorl-thumbnail, je fais directement un image.crop((0, 0, 150, 150)) via pillow, ça m'irait de ne rien faire et de simplement refuser les images plus grandes que 150x150 (et je remarque une API Image.thumbnail() qui n'a pas l'air mal non plus).

On est d'accord sur le fait que ça signifie qu'un utilisateur normal n'aura ainsi jamais que le vague et insignifiant coin supérieur gauche de son image "trop grande" ?

#95 - 23 octobre 2018 18:13 - Benjamin Dauvergne

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Vu avec Stéphane:

- c'est limité en dur à 200x200 pas de cropping
- les tests valident cela

#96 - 23 octobre 2018 18:27 - Frédéric Péters

Et sur le partage lors du SSO, j'ai bien lu que tu disais à Paul, "plus tard", mais je note quand même,

En oidc, qui sera le premier usage Strasbourg, il me semble que ça va péter façon :

```
TypeError: &lt;authentic2.attribute_kinds.ProfileImageFile object at 0x7f8faf463690&gt; is not JSON serializab
le
```

Et en SAMLv2, là j'ai testé, et ça produit :

```
&lt;saml:Attribute Name="photo" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" FriendlyName=""
&gt;&lt;saml:AttributeValue&gt;&lt;authentic2.attribute_kinds.ProfileImageFile object at 0x7f0020ae8ed0&gt;&lt
;/saml:AttributeValue&gt;&lt;/saml:Attribute&gt;
```

#97 - 23 octobre 2018 18:28 - Frédéric Péters

> c'est limité en dur à 200x200 pas de cropping

Ça rend cela inutilisable pour les usagers. En mode "pas envie, je laisse quelqu'un faire ça plus tard", ok, mais ça ne me semble même pas être le cas ?

#98 - 23 octobre 2018 18:32 - Benjamin Dauvergne

Moi on m'a dit API on m'a jamais dit SSO, je regarde.

#99 - 23 octobre 2018 18:38 - Frédéric Péters

> (...) l'idée d'avoir un nouveau type d'attribut pour uploader un fichier, stocker ce fichier et servir l'URL publique du fichier dans les attributs de SSO.

C'est dans la description du ticket.

#100 - 23 octobre 2018 19:16 - Benjamin Dauvergne

- Fichier 0001-support-avatar-picture-in-user-profile-26022.patch ajouté

Avec le support au SSO testé en OIDC et SAML, je n'ai pas fait CAS mais a priori c'est bon vu que request est bien dans le contexte.

#101 - 23 octobre 2018 20:51 - Frédéric Péters

Super, reste quand même la question importante de l'évolution, est-il envisagé que tu acceptes qu'authentic fasse le boulot de redimensionner de manière "intelligente" l'image fournie, travail qui pourrait passer par sorl.thumbnail ? (sorl.thumbnail parce que ça permet très facilement quelque chose de plus fin que juste "découper un carré au milieu", qu'on peut lui dire de favoriser le haut de l'image, ce qui la plupart du temps donnera un meilleur résultat, etc.)

#102 - 24 octobre 2018 07:26 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Super, reste quand même la question importante de l'évolution, est-il envisagé que tu acceptes qu'authentic fasse le boulot de redimensionner de manière "intelligente" l'image fournie, travail qui pourrait passer par sorl.thumbnail ? (sorl.thumbnail parce que ça permet très facilement quelque chose de plus fin que juste "découper un carré au milieu", qu'on peut lui dire de favoriser le haut de l'image, ce qui la plupart du temps donnera un meilleur résultat, etc.)

Je n'aime pas sorl-thumbnail, c'est une abstraction inutile, ici j'étais bien incapable de l'appeler et gérer ensuite l'image moi même (ça fait des trucs avec un key-value store dans l'idée que c'est simplement un cache de la vrai image ce qui n'est pas le cas ici) mais parvenir à reproduire un crop=center avec du PIL je ne pense pas que ça prenne plus d'une dizaine de ligne de code même si je pense que ce sera inutile à partir du moment où on précise la taille de l'image, à la rigueur il faudrait aller jusqu'à ce que propose la plupart des réseaux sociaux i.e. avoir un widget d'upload qui permette le cropping manuellement en affichant un cadre à déplacer sur l'image; mais bon beaucoup de boulot pour un truc encore plus inutile.

Vraiment je ne veux pas introduire de complexité que je ne comprenne pas pour une fonctionnalité qui n'a aucun intérêt et sert simplement à faire plaisir à un client (c'est comme le JSON attaché à un mail, ça ne va servir qu'à Strasbourg...).

#103 - 24 octobre 2018 07:36 - Frédéric Péters

> mais parvenir à reproduire un crop=center avec du PIL je ne pense pas que ça prenne plus d'une dizaine de ligne de code même si je pense que ce sera inutile à partir du moment où on précise la taille de l'image.

Les usagers n'ont souvent aucune idée de ce qu'est la taille d'une image, et ne sont certainement pas en mesure de redimensionner.

> Vraiment je ne veux pas introduire de complexité que je ne comprenne pas pour une fonctionnalité qui n'a aucun intérêt et sert simplement à faire plaisir à un client (c'est comme le JSON attaché à un mail, ça ne va servir qu'à Strasbourg...).

Les maquettes du nouveau portail agent reprennent un avatar pour l'agent.

#104 - 24 octobre 2018 07:59 - Pierre Cros

Benjamin Dauvergne a écrit :

> une fonctionnalité qui n'a aucun intérêt et sert simplement à faire plaisir à un client

On a changé d'ère, parce que Publik a beaucoup progressé. Les détails de ce type apportent aujourd'hui une vraie plus value. Les grosses fonctionnalités sont là pour l'essentiel, mais on a pas mal de boulot sur la façon de les présenter, sur la simplification et la joliesse. L'avatar ça s'inscrit dans cette logique là, histoire de montrer qu'un logiciel pour les collectivités ne doit pas nécessairement avoir une interface austère ("ne pas abandonner l'innovation et la modernité à Google et Facebook" disais-je en conclusion d'un forum Adullact, sous les vivas de la foule).

#105 - 24 octobre 2018 09:19 - Frédéric Péters

fwiw, copie du code de redimensionnement que j'utilisais dans un autre projet, avant de basculer sur sorl.thumbnail,

```
        image = Image.open(filename)
        if abs((1.0*width/height) - (1.0*image.size[0]/image.size[1])) > 0.1:
            # aspect ratio change, crop the image first
            box = [0, 0, image.size[0], int(image.size[0] * (1.0*height/width))]

            if box[2] > image.size[0]:
                box = [int(t*(1.0*image.size[0]/box[2])) for t in box]
            if box[3] > image.size[1]:
                box = [int(t*(1.0*image.size[1]/box[3])) for t in box]

            if image.size[0] > image.size[1]:  # landscape
```

```
                box[0] = (image.size[0] - box[2]) / 2  # keep the middle
                box[2] += box[0]
            else:
                box[1] = (image.size[1] - box[3]) / 4  # keep mostly the top
                box[3] += box[1]

            image = image.crop(box)
        image = image.resize([width, height], Image.ANTIALIAS)
```

#106 - 24 octobre 2018 12:22 - Benjamin Dauvergne

Stop au spam, j'attends validation du code pour l'avoir en recette puisque c'est urgentissime.

#107 - 24 octobre 2018 12:40 - Pierre Cros

Fred est en congés. Qui doit valider ?

#108 - 24 octobre 2018 13:14 - Frédéric Péters

Sans redimensionnement c'est inutilisable pour l'usager, ceci est ma relecture.

#109 - 29 octobre 2018 15:57 - Frédéric Péters

> Sans redimensionnement c'est inutilisable pour l'usager, ceci est ma relecture.

J'ai donc créé #27644 avec un patch pour redimensionner.

#110 - 30 octobre 2018 11:32 - Frédéric Péters

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

```
commit a5d652ce81436f1f3fab0a66680da86c4879fe03
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Tue Sep 4 16:26:15 2018 +0200

    support avatar picture in user profile (#26022)
```

#111 - 13 décembre 2018 22:40 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

screenshot_avatar_photo00.png	45 ko	03 septembre 2018	Paul Marillonnet
0001-WIP-support-avatar-picture-in-user-profile-26022.patch	3,12 ko	04 septembre 2018	Paul Marillonnet
0001-WIP-support-avatar-picture-in-user-profile-26022.patch	7,01 ko	04 septembre 2018	Paul Marillonnet
0001-WIP-support-avatar-picture-in-user-profile-26022.patch	10 ko	05 septembre 2018	Paul Marillonnet
account_avatar.png	118 ko	05 septembre 2018	Paul Marillonnet
0001-WIP-support-avatar-picture-in-user-profile-26022.patch	13,4 ko	11 septembre 2018	Paul Marillonnet
0001-WIP-support-avatar-picture-in-user-profile-26022.patch	15,8 ko	12 septembre 2018	Paul Marillonnet
0001-WIP-support-avatar-picture-in-user-profile-26022.patch	30,5 ko	17 septembre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	1,75 Mo	18 septembre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	1,74 Mo	18 septembre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	1,75 Mo	18 septembre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	1,75 Mo	19 septembre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	28,9 ko	19 septembre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	28,9 ko	19 septembre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	28,8 ko	19 septembre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	29,1 ko	19 septembre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	29,1 ko	01 octobre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	23,6 ko	04 octobre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	21,9 ko	04 octobre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	19,8 ko	11 octobre 2018	Paul Marillonnet

0001-support-avatar-picture-in-user-profile-26022.patch	20,6 ko	12 octobre 2018	Paul Marillonnet
0001-add-an-html_value-method-to-attribut-kinds.patch	4,71 ko	22 octobre 2018	Benjamin Dauvergne
0001-support-avatar-picture-in-user-profile-26022.patch	20,1 ko	22 octobre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	19,8 ko	22 octobre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	19,5 ko	22 octobre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	19,3 ko	22 octobre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	19,1 ko	22 octobre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	19 ko	23 octobre 2018	Paul Marillonnet
0001-support-avatar-picture-in-user-profile-26022.patch	17,9 ko	23 octobre 2018	Benjamin Dauvergne
0001-support-avatar-picture-in-user-profile-26022.patch	18,1 ko	23 octobre 2018	Benjamin Dauvergne
0001-support-avatar-picture-in-user-profile-26022.patch	29,6 ko	23 octobre 2018	Benjamin Dauvergne