

## w.c.s. - Development #39517

### gestion d'un stockage de fichier distant

04 février 2020 11:39 - Thomas Noël

<b>Statut:</b>	Fermé	<b>Début:</b>	04 février 2020
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Thomas Noël	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
Possibilité de déclarer dans site-options.cfg des systèmes de stockage des fichiers uploadés, sous la forme :			
<pre>[storage-foo] label = stockage sur foo class = wcs.qommon.upload_storage.RemoteFile ws = https://crypto.example.net/ws/</pre>			
Sur chaque champ fichier on dispose alors d'un choix de stockage "stockage sur foo".			
class est la classe d'objet, dont le constructeur recevra la description du stockage (label, class, ws).			
On disposera d'une classe <code>wcs.qommon.upload_storage.RemoteOpaqueUploadStorage</code> , pour laquelle <code>ws</code> est un webservice qui va recevoir le fichier en json (filename/content_type/base64content) et renvoie une URL vers le stockage distant de ce fichier (cf <a href="#">#39431</a> )			
<b>Demandes liées:</b>			
Précède w.c.s. - Development #39976: gestion des fichiers distants sur les do...		<b>Fermé</b>	<b>05 février 2020</b>

#### Révisions associées

##### Révision 36710b94 - 20 février 2020 00:39 - Thomas Noël

add alternative storage system (#39517)

#### Historique

##### #1 - 04 février 2020 11:45 - Thomas Noël

Un travail d'implémentation est visible sur <http://git.entrouvert.org/wcs.git/?h=wip/alternate-file-storage>

Les objets `PickleableUpload` gagnent un attribut dictionnaire "storage\_attrs" quand ils sont passés par `RemoteOpaqueUploadStorage`. Dans ce dictionnaire on a un `redirect_url` qui est l'URL vers laquelle rediriger l'utilisateur quand il veut ouvrir le fichier ; cette URL de redirection est signée au moment de clic, permettant un accès sécurisé à un système distant d'accès au fichier.

En export JSON, les fichiers distants ont un content vide mais on renvoie `storage_attrs`.

##### #2 - 05 février 2020 12:39 - Thomas Noël

- Fichier `0001-add-alternative-storage-system-39517.patch` ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Voici donc un premier patch, avec un test assez simple mais qui correctement le code. Ce qui est important c'est aussi que ça ne casse rien pour les sites qui n'ont aucune section `[storage-xxx]` dans leur site-options, ça permet ainsi de considérer cette évolution comme "bêta".

Ce qui manque pour l'instant mais que je ne juge pas prioritaire à ce stade :

- la gestion des fichiers déposés dans une action "joindre un fichier"
- la gestion de `PickleableUpload.receive` pour envoyer du contenu dans un fichier existant (typiquement pour générer un fichier dans un workflow) -- ça va juste salement cracher, mais modifier ça me semble rendre le patch moins facile à lire
- la gestion des `UploadStorageError` (par exemple quand le webservice va planter), je ne sais pas trop encore quoi en faire, là aussi je pense que ça pourrait venir dans un patch suivant de "solidification"
- ne pas permettre de voir les fichiers en frontoffice (ni lien ni redirection), via une option à ajouter dans `RemoteOpaqueUploadStorage` à côté de

Ce qui me chiffonne un peu dans ma solution c'est les tests du genre « if hasattr(upload, 'storage\_attrs') » pour savoir si un fichier est distant ou pas. Peut-être qu'on pourrait avoir des attributs plus explicites, genre upload.opaque ou upload.remote autre... mais ça va pas forcément rendre les choses plus simples. Idées bienvenues le cas échéant. (Dans le patch original de Frédéric il y avait un test sur un attribut upload.redirect\_url)

### #3 - 05 février 2020 13:01 - Frédéric Péters

Ce qui me chiffonne un peu dans ma solution c'est les tests du genre « if hasattr(upload, 'storage\_attrs') » pour savoir si un fichier est distant ou pas. Peut-être qu'on pourrait avoir des attributs plus explicites, genre upload.opaque ou upload.remote autre... mais ça va pas forcément rendre les choses plus simples. Idées bienvenues le cas échéant. (Dans le patch original de Frédéric il y avait un test sur un attribut upload.redirect\_url)

Je verrais un truc comme upload.can\_thumbnail(), upload.get\_json\_value() (pour les deux vus rapidement); et dans Upload alors la mécanique qui va déléguer la question à la classe de stockage.

### #4 - 05 février 2020 14:59 - Thomas Noël

- Fichier 0001-upload\_storage-add-can\_thumbnails-method-on-Picklabl.patch ajouté

Frédéric Péters a écrit :

Je verrais un truc comme upload.can\_thumbnail()

Je viens d'ajouter un patch dans la branche pour gérer ça, ci-joint. Il reste que je n'arrive pas à cerner si on reçoit toujours bien des PicklableUpload et pas des Upload, alors je me retrouve à faire des conditions lourdes comme « if hasattr(upload, 'can\_thumbnail') and upload.can\_thumbnail() » ... est-ce que je me fais peur à moi-même ?

En tout cas si c'est bien l'idée que tu avais exprimée, je vais ajouter ensuite get\_json\_value() et un receive().

### #5 - 05 février 2020 16:00 - Frédéric Péters

est-ce que je me fais peur à moi-même ?

Lance les tests sans les hasattr pour voir ?

### #6 - 05 février 2020 17:16 - Thomas Noël

Frédéric Péters a écrit :

est-ce que je me fais peur à moi-même ?

Lance les tests sans les hasattr pour voir ?

```
py3-pylint-coverage: commands succeeded
congratulations :)
```

Courageux, j'ai donc refais le commit sans les hasattr. C'est dans la branche, voici l'interdiff :

```
diff --git a/wcs/fields.py b/wcs/fields.py
index 50583408..ebaf5ea4 100644
--- a/wcs/fields.py
+++ b/wcs/fields.py
@@ -1041,7 +1041,7 @@ class FileField(WidgetField):
     t = TemplateIO(html=True)
     t += htmltext('<div class="file-field">')
     t += htmltext('<a download="%s" href="[download]?f=%s">' % (value.base_filename, self.id)
-    if include_image_thumbnail and hasattr(value, 'can_thumbnail') and value.can_thumbnail():
+    if include_image_thumbnail and value.can_thumbnail():
         t += htmltext('' % self.id)
     t += htmltext('<span>%s</span>') % value
     t += htmltext('</a></div>')
```

```
diff --git a/wcs/forms/common.py b/wcs/forms/common.py
index af4486c3..4b9214e2 100644
--- a/wcs/forms/common.py
+++ b/wcs/forms/common.py
```

```

@@ -70,7 +70,7 @@ class FileDirectory(Directory):
     response = get_response()

     if self.thumbnails:
-         if hasattr(file, 'can_thumbnail') and file.can_thumbnail():
+         if file.can_thumbnail():
             try:
                 thumbnail = misc.get_thumbnail(file.get_filename(),
                                                content_type=file.content_type)
@@ -686,7 +686,7 @@ class FormStatusPage(Directory, FormTemplateMixin):
     file_url = 'files/%s/' % fn

     if get_request().form.get('thumbnail') == '1':
-         if hasattr(file, 'can_thumbnail') and file.can_thumbnail():
+         if file.can_thumbnail():
             file_url += 'thumbnail/'
         else:
             raise errors.TraversalError()

```

Ca te semble une façon correcte de faire ? (ajout du can\_thumbnail):

<http://git.entrouvert.org/wcs.git/commit/?h=wip/39517-alternate-file-storage&id=c81c1bbc02907fce1b67996c4a31e2d70b46b56a>

#### #7 - 05 février 2020 17:19 - Frédéric Péters

Yep ça me parait bien.

Ça me fait voir quand même dans le commit cet autre bout avec un hasattr qui serait à remplacer par une méthode :

```

+         if hasattr(file, 'storage_attrs'): # remote storage
             if not file.storage_attrs.get('redirect_url'):

```

#### #8 - 06 février 2020 00:06 - Thomas Noël

- Statut changé de Solution proposée à En cours

Frédéric Péters a écrit :

Yep ça me parait bien.

Ça me fait voir quand même dans le commit cet autre bout avec un hasattr qui serait à remplacer par une méthode

Yep. Je pense pour cela créer `upload.has_redirect_url()` et si True, `upload.get_redirect_url(backoffice=True/False)` qui pourrait ne rien renvoyer, et dans ce cas ni lien ni redirection. L'absence d'url serait décidée via un « {front,back}office\_redirect = true/false » dans le [storage-xxx]. Le cas d'usage CNIL serait qu'on n'affiche pas de lien en front.

#### #9 - 06 février 2020 16:55 - Thomas Noël

- Fichier 0001-upload\_storage-add-frontoffice-backoffice-link-prese.patch ajouté

- Statut changé de En cours à Solution proposée

La suite donc qui implémente ce `has_redirect_url/get_redirect_url` afin d'afficher ou pas des liens en frontoffice ou backoffice sur des fichiers distants. Au passage un test à montré qu'un file pouvait être un `AttachmentEvolutionPart` et donc j'y ai ajouté ces méthodes.

(Je vais continuer sur le `get_json_value` et arrêter cette branche ensuite, la "vraie" relecture peut donc déjà commencer, je pense)

#### #10 - 07 février 2020 15:31 - Thomas Noël

- Fichier 0001-upload\_storage-add-get\_json\_value-and-use-it-in-json.patch ajouté

Je viens de pousser l'ajout de `get_json_value` dans la branche (j'attache le patch, à toutes fins utiles).

La limite actuelle de l'implémentation reste la gestion des `receive()` et donc ce qui concerne l'upload de document via des formulaires de workflows et leur transfert dans les données de traitement, par exemple. Je pense cependant en faire un ticket séparé, c-à-d considérer que le présent ticket couvre la notion de "stockage distant sur les demandes", une version bêta. Ce qui me semble nécessaire c'est de valider puis envoyer ça en recette pour "faire tourner" afin de se rassurer "en live" qu'il n'y a pas de régression sur le `PickleableUpload` natif.

#### #11 - 18 février 2020 09:19 - Frédéric Péters

La limite actuelle de l'implémentation reste la gestion des `receive()` et donc ce qui concerne l'upload de document via des formulaires de workflows et leur transfert dans les données de traitement, par exemple. Je pense cependant en faire un ticket séparé.

Ok mais faut ce ticket.

~~

Il va falloir rebaser en faisant attention, notamment à ce bout :

```
+ def __setstate__(self, dict):  
+     self.__dict__.update(dict)  
+     if hasattr(self, 'data'):  
+         # backward compatibility with older w.c.s. version  
+         self.fp = StringIO(self.data)  
+     del self.data
```

(qui est BytesIO dans master).

~~

À repasser sur le branche j'aurais déplacé la partie "RemoteOpaqueUploadStorage" dans un commit séparé/final mais ok.

~~

```
+ def get_redirect_url(self, upload, backoffice=False):  
+     raise NotImplementedError('no get_redirect_url on AttachmentEvolutionPart object')
```

Sans doute un ticket également à faire pour gérer ça.

~~

```
+     if hasattr(self, 'storage_attrs'): # remote storage
```

Je serais pour `s/remote storage/alternative storage/`, même si on n'a pas de perspective particulière à ce sujet.

#### #12 - 18 février 2020 12:27 - Thomas Noël

- Précède *Development #39976*: gestion des fichiers distants sur les données de traitement (`upload.receive()`) ajouté

#### #13 - 18 février 2020 12:35 - Thomas Noël

Frédéric Péters a écrit :

La limite actuelle de l'implémentation reste la gestion des `receive()` et donc ce qui concerne l'upload de document via des formulaires de workflows et leur transfert dans les données de traitement, par exemple. Je pense cependant en faire un ticket séparé.

Ok mais faut ce ticket.

Hop, [#39976](#)

Il va falloir rebaser en faisant attention, notamment à ce bout :

C'est rebasé avec ce BytesIO.

À repasser sur le branche j'aurais déplacé la partie "RemoteOpaqueUploadStorage" dans un commit séparé/final mais ok.

Ouf (je suis pas assez doué en rebase pour ça)

```
raise NotImplementedError('no get_redirect_url on AttachmentEvolutionPart object')
```

Sans doute un ticket également à faire pour gérer ça.

Tu veux dire gérer des fichiers distants au niveau des AttachmentEvolutionPart ?

Je serais pour `s/remote storage/alternative storage/`, même si on n'a pas de perspective particulière à ce sujet.

Yep c'est changé en "alternative storage" (sur la branche).

#### #14 - 18 février 2020 12:50 - Frédéric Péters

```
raise NotImplementedError('no get_redirect_url on AttachmentEvolutionPart object')
```

Sans doute un ticket également à faire pour gérer ça.

Tu veux dire gérer des fichiers distants au niveau des AttachmentEvolutionPart ?

Pas la moindre idée je suis juste parti sur l'idée que ce NotImplementedError était uniquement temporaire, en attendant "quelque chose". S'il a vocation à rester là je serais plutôt pour taper une autre exception, pour laquelle on recevra une trace si jamais le code se trouve appelé.

#### #15 - 18 février 2020 15:06 - Thomas Noël

Nouvelle branche poussée avec des AssertionError à la place des NotImplementedError (pas d'URL distante pour les fichiers locaux, ni pour les fichiers enregistrés dans l'historique -- du moins pour ces derniers pas dans cette version où has\_redirect\_url est toujours False).

#### #16 - 19 février 2020 20:11 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

```
+ 'content': force_text(base64.b64encode(upload.get_content())),
```

Détail mais on a désormais un get\_base64\_content() qui pourrait être utilisé ici.

Ok avec ou sans ce changement, en tapant tout dans un unique commit.

Je pense aussi que ce serait utile de créer un ticket concernant la gestion des erreurs, là il me semble que les UploadStorageError vont juste donner une série d'erreurs 500. (mais avant même ça, ici, changer les "remote storage returns" en "remote storage returned").

#### #17 - 20 février 2020 00:46 - Thomas Noël

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

Frédéric Péters a écrit :

```
+ 'content': force_text(base64.b64encode(upload.get_content())),
```

Détail mais on a désormais un get\_base64\_content() qui pourrait être utilisé ici.

J'avais tenté mais ça finissait par une mini erreur sur test\_formdef\_submit\_with\_varname :

```
> assert formdef.fields[4].get_json_value(data)[k] == payload['data']['file'][k]
E AssertionError: assert 'dGVzdA==\n' == 'dGVzdA=='
```

parce que base64.b64encode d'un côté et base64.encodestring de l'autre. Je n'avais pas voulu tripoter ça ici (bien que je pense que c'est sans impact).

Ok avec ou sans ce changement, en tapant tout dans un unique commit.

Je pense aussi que ce serait utile de créer un ticket concernant la gestion des erreurs, là il me semble que les UploadStorageError vont juste donner une série d'erreurs 500. (mais avant même ça, ici, changer les "remote storage returns" en "remote storage returned").

Ok pour returned.

Et oui pour la gestion des erreurs, j'ai posé [#40044](#)

```
commit 36710b94c870946fb83019e0fb7690c937fe4021
Author: Thomas NOEL <tnoel@entrouvert.com>
Date: Wed Feb 5 12:28:36 2020 +0100
```

```
add alternative storage system (#39517)
```

#### #18 - 20 février 2020 02:14 - Frédéric Péters

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

## Fichiers

---

0001-add-alternative-storage-system-39517.patch	29,4 ko	05 février 2020	Thomas Noël
0001-upload_storage-add-can_thumbnails-method-on-Picklabl.patch	5,96 ko	05 février 2020	Thomas Noël
0001-upload_storage-add-frontoffice-backoffice-link-prese.patch	12,2 ko	06 février 2020	Thomas Noël
0001-upload_storage-add-get_json_value-and-use-it-in-json.patch	5,4 ko	07 février 2020	Thomas Noël