

Passerelle - Development #36025

cart@ds, dépôt des pièces complémentaires

11 septembre 2019 11:19 - Frédéric Péters

Statut:	Fermé	Début:	11 septembre 2019
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Après [#36024](#) on aura obtenu dans le connecteur les pièces complémentaires, il faudra ensuite les transmettre à cart@ds.

4.2 de leur documentation, Dépôt des pièces :

Chaque pièce doit obligatoirement s'appeler `<idPiece>-<codePiece><numFichier>.<extension>` (ex : pour le plan de situation du terrain d'un PCMI, 1-PC011.pdf et 1-PC012.pdf).

`<idPiece>` correspond à l'identifiant de la pièce.
`<codePiece>` correspond au code la pièce.
`<numFichier>` correspond au numéro du fichier pouvant aller jusqu'à 6.
`<extension>` correspond au type du fichier.

Paramètres en entrée :

request : objet contenant les informations sur le fichier envoyé.

Signature de la méthode :

```
/// <summary>
/// Fonction d'upload
/// </summary>
/// <param name="request">Informations de la requête</param>
void UploadFile(RemoteFileInfoUploadRequest request);
```

Exemple d'appel au service web pour télécharger un fichier sur le serveur :

```
this.ServiceDocumentation.UploadFile(new RemoteFileInfoUploadRequest
{
  IdDossier = idDossier,
  Token = token,
  NomFichier = nomFichier,
  Length = file.ContentLength,
  FileByteStream = file.InputStream,
  InformationsComplementaires = new Dictionary<string, string>()
  {
    { "idDosPiece", idDosPiece }
  }
});
```

Le paramètre IdDossier correspond à l'identifiant du dossier dans Cart@DS CS.

Le paramètre NomFichier correspond au nom du fichier déposé (respect de la règle de nommage).

Le paramètre Length correspond à la taille du contenu du fichier.

Le paramètre FileByteStream correspond au contenu du fichier au format Stream.

Le paramètre InformationsComplementaires est un dictionnaire de données dans lequel il est possible d'ajouter des paramètres supplémentaires. Voici la liste des paramètres disponibles :

- * arborescence : nom du répertoire dans lequel déposer le fichier.
- * idDosPiece : identifiant de la pièce du dossier concernée.

* idEtapeDossier : identifiant d'une étape qui sera associée au document. Attention, fonctionne uniquement sous SharePoint ou Alfresco.

Nous n'avons pas de retour de la part de la méthode.

Révisions associées

Révision bbb86c3b - 13 septembre 2019 13:24 - Frédéric Péters

cartads_cs: add endpoint and job to send additional files to cartads (#36025)

Historique

#1 - 13 septembre 2019 09:21 - Frédéric Péters

- Fichier 0001-cartads_cs-add-endpoint-and-job-to-send-additional-f.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Donc après avoir transféré une série de fichiers, ici un endpoint qui crée juste le job pour l'envoi, pour rendre la main rapidement, puis le job.

#2 - 13 septembre 2019 10:22 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Pour le "assert resp is None", il se produit dans quel cas ? Si c'est un cas "possible", on pourrait y ajouter un petit texte.

Pour le reste c'est ok. Resterait la question du suivi de l'envoi ; par exemple un endpoint qui listerait les documents en attente (sent_to_cartads==None) qui permettrait de savoir si tout s'est bien passé ? (Je dis ça sans savoir exactement à quel endroit l'envoi aura lieu)

#3 - 13 septembre 2019 10:32 - Benjamin Dauvergne

Thomas Noël a écrit :

Pour le "assert resp is None", il se produit dans quel cas ? Si c'est un cas "possible", on pourrait y ajouter un petit texte.

Je crois que ça n'arrive jamais mais que Fred a envie d'être au courant si le WSDL change et que ça se met à renvoyer quelque chose.

Pour le reste c'est ok. Resterait la question du suivi de l'envoi ; par exemple un endpoint qui listerait les documents en attente (sent_to_cartads==None) qui permettrait de savoir si tout s'est bien passé ? (Je dis ça sans savoir exactement à quel endroit l'envoi aura lieu)

De mon côté j'ai un souci avec l'utilisation de `i` pour construire le nom du fichier qui va dépendre du retour de `CartaDSFile.objects.filter(..., sent_to_cartads__isnull=True)` si on a deux fichiers du même type que le deuxième foire puis est rejoué on aura le même `i`, il faudrait toujours lister toutes les pièces dans un ordre croissant (genre via `pk`) et filtrer dans la boucle en fonction de `sent_to_cartads`.

#4 - 13 septembre 2019 10:33 - Benjamin Dauvergne

- Statut changé de Solution validée à En cours

#5 - 13 septembre 2019 13:27 - Frédéric Péters

- Fichier 0001-cartads_cs-add-endpoint-and-job-to-send-additional-f.patch ajouté
- Statut changé de En cours à Solution proposée

Voilà avec `ordering=['id']` posé; et la vérification du `sent_to_cartads` sortie du `.filter()` pour aller dans la boucle.

Pour le "assert resp is None", il se produit dans quel cas ? Si c'est un cas "possible", on pourrait y ajouter un petit texte.

Non ce n'est pas un cas possible, ils écrivent « Nous n'avons pas de retour de la part de la méthode. » dans la documentation; c'est comme l'écrit Benjamin juste un bout pour être alerté si jamais quelque chose change à ce niveau.

#6 - 13 septembre 2019 14:44 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Ok pour moi.

#7 - 13 septembre 2019 15:06 - Frédéric Péters

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

```
commit bbb86c3bad727804ce4a7c5c0994ac08dac7a155
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Fri Sep 13 08:44:52 2019 +0200
```

```
cartads_cs: add endpoint and job to send additional files to cartads (#36025)
```

#8 - 16 septembre 2019 14:15 - Frédéric Péters

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

Fichiers

0001-cartads_cs-add-endpoint-and-job-to-send-additional-f.patch	7,17 ko	13 septembre 2019	Frédéric Péters
0001-cartads_cs-add-endpoint-and-job-to-send-additional-f.patch	7,45 ko	13 septembre 2019	Frédéric Péters