

Passerelle - Development #35325

developper le connecteur Gesbac

09 août 2019 10:58 - Serghei Mihai

Statut:	Fermé	Début:	09 août 2019
Priorité:	Normal	Echéance:	
Assigné à:	Serghei Mihai	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
En résumé: création des fichiers CSV avec les demandes, envoi vers un SFTP distant, lecture des fichiers réponse depuis le SFTP distant. Doc fournie par l'éditeur via la métropole de Grenoble.			
Demandes liées:			
Lié à Passerelle - Development #36133: jsonschema: ajouter la prise des valeu...			Rejeté 16 septembre 2019

Révisions associées

Révision bf743017 - 23 septembre 2019 09:53 - Serghei Mihai

gesbac: initial connector (#35325)

Historique

#1 - 16 septembre 2019 10:36 - Serghei Mihai

- Lié à Development #36133: jsonschema: ajouter la prise des valeurs par défaut ajouté

#2 - 16 septembre 2019 12:37 - Benjamin Dauvergne

De ce que je lis dans le .docx les transferts SFTP sont gérés par la DSI, donc on a juste à produire des CSV dans un le répertoire media/gesbac/<slug>/publik_vers_gesbac/ en gros.

#3 - 16 septembre 2019 12:59 - Serghei Mihai

Non, nous devons envoyer les fichiers sur un SFTP de la DSI.
Ensuite Gesbac vient chercher (je ne sais pas comment) les fichiers dans ce répertoire, il les archive, etc.

#4 - 16 septembre 2019 13:12 - Benjamin Dauvergne

Serghei Mihai a écrit :

Non, nous devons envoyer les fichiers sur un SFTP de la DSI.
Ensuite Gesbac vient chercher (je ne sais pas comment) les fichiers dans ce répertoire, il les archive, etc.

Alors ça doit être une interprétation de ça dans le .docx :

Les transferts des fichiers csv Publik vers le serveur SFPT seront initiés par un séquenceur DSI le soir après la fin de la saisie des validations des dossiers d'instruction de demande de badge.

#5 - 16 septembre 2019 16:17 - Benjamin Dauvergne

Peut-être que ça passera par deux FTPs mais ce serait bien d'avoir confirmation dès maintenant avant de développer du fonctionnement final.

#6 - 16 septembre 2019 16:23 - Serghei Mihai

Il y a un seul SFTP (#35849) sur lequel nous déposons les fichiers.
Ensuite la DSI ou Gesbac se débrouillent pour les récupérer et les donner à manger au logiciel métier, archiver les fichiers posés.

C'est ce qu'on a convenu lors d'une réunion téléphonique.

#7 - 16 septembre 2019 16:47 - Benjamin Dauvergne

Serghei Mihai a écrit :

Il y a un seul SFTP (#35849) sur lequel nous déposons les fichiers.
Ensuite la DSI ou Gesbac se débrouillent pour les récupérer et les donner à manger au logiciel métier, archiver les fichiers posés.

C'est ce qu'on a convenu lors d'une réunion téléphonique.

Leur demander de mettre à jour leur document, je suis désolé d'être tatasse mais vraiment c'est mieux quand ils écrivent noir sur blanc ce qu'ils vont faire/veulent qu'on fasse.

Concernant le SFTP à voir si tu préfères gérer ça hors passerelle (genre un énième script client) ou via le nouveau champ sftp développé pour MDEL (je pense que c'est mieux d'avoir le suivi des transferts dans les logs passerelle, ça fait gagner du temps).

#8 - 16 septembre 2019 17:01 - Serghei Mihai

- Assigné à mis à Serghei Mihai

J'utilise le nouveau champ SFTP que tu as développé pour MDEL, ça marche très bien.

Pour logger les appels du connecteur je rajoute une table supplémentaire qui me permettra de stocker et timestamp-er les données envoyées depuis la demande.

L'idée est aussi de stocker les réponses de Gesbac dans la table. Lancer la recherche des fichiers réponse sur le SFTP toutes les heures (via hourly), stocker leur contenu et le servir lors d'un check par Publik du statut de la demande.

J'ai poussé la wip.

Pour la même demande il peut y avoir plusieurs échanges entre Publik et Gesbac, genre Gesbac indique via son fichier de réponse qu'il manque un paramètre.

Dans ce cas un nouveau fichier CSV est déposé sur le SFTP pour la même demande.

#9 - 17 septembre 2019 09:54 - Serghei Mihai

- Fichier 0001-gesbac-initial-connector-35325.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Premier patch avec création et lecture des fichiers CSV sur le SFTP.

Dans les suffixes xxxxxx des noms des fichiers je met le numéro de la demande et un numéro de séquence pour pouvoir différencier les potentiels aller-retours entre les demandes Publik et Gesbac.

#10 - 17 septembre 2019 10:53 - Frédéric Péters

```
for param in ['demand_date', 'demand_time', 'producer_code'...
```

Plutôt que répéter cette liste, tirer ça du schéma ?

~

Mais avant ça peut-être, j'essaie de comprendre le fond :

- Une demande est envoyée de w.c.s. au connecteur, celui-ci l'enregistre sous forme de deux lignes dans un fichier CSV sur un serveur SFTP distant.
- Quand il est demandé la réponse à dune demande, il se passe un truc avec du SFTP entrant et des fichiers CSV qui sont cherchés et un objet CSV qui est créé et un "data" qui est une liste (des réponses?).

~

Cet objet "FileHandler", en soit, il apporte quelque chose ? Il me semble que non et qu'il est plus commun d'avoir ces opérations de traitement dans la classe du connecteur, pas dans une classe intermédiaire qui fait rien.

#11 - 17 septembre 2019 11:47 - Benjamin Dauvergne

Ouais ça va pas du tout, on peut pas gérer un fichier CSV distant comme ça, l'objectif à la lecture du .docx est clairement pas d'avoir du temps réel (ça dit clairement la DSI, tous les soirs, etc...).

J'aurai vu un truc en 2 temps :

- à chaque appel send:
 1. on cherche ou on crée un objet CSV dans un statut "open"
 2. on crée un objet CsvLine avec les données du formulaire, fini (aucun souci de concurrence possible, tout va bien)
- tous les n-période (tous les soirs disont) on prend le fichier CSV open le plus vieux, on le passe en "closed", ensuite on se connecte au FTP on pose un CSV avec l'identifiant incrémental du CSV (genre le .id ça doit suffir), quand c'est terminé on passe l'objet CSV en statut "sent"

Si jamais il était besoin de conserver l'état d'un formulaire pour les retours alors faut un objet FormData qu'on pourra relier à plusieurs CSV sortant ou

rentrant.

Terminé.

#12 - 17 septembre 2019 11:51 - Benjamin Dauvergne

Aussi si on vise un seul sftp je pense qu'un seul champ SFTPField suffit, chemin et préfixe peuvent être mis en dur (on parle d'un connecteur contrib/ après tout).

#13 - 17 septembre 2019 12:11 - Serghei Mihai

Benjamin Dauvergne a écrit :

Ouais ça va pas du tout, on peut pas gérer un fichier CSV distant comme ça, l'objectif à la lecture du .docx est clairement pas d'avoir du temps réel (ça dit clairement la DSI, tous les soirs, etc...).

J'aurai vu un truc en 2 temps :

- à chaque appel send:
 1. on cherche ou on crée un objet CSV dans un statut "open"
 2. on crée un objet CsvLine avec les données du formulaire, fini (aucun souci de concurrence possible, tout va bien)
- tous les n-période (tous les soirs disont) on prend le fichier CSV open le plus vieux, on le passe en "closed", ensuite on se connecte au FTP on pose un CSV avec l'identifiant incrémental du CSV (genre le .id ça doit suffir), quand c'est terminé on passe l'objet CSV en statut "sent"

Faire cela tous les n-périodes ou au moment de l'appel webservice depuis wcs, je ne vois pas différence. Que la DSI (ou Gesbac) vienne récupérer les fichiers tous les soirs, toutes les n heures, c'est leur affaire.

Si jamais il était besoin de conserver l'état d'un formulaire pour les retours alors faut un objet FormData qu'on pourra relier à plusieurs CSV sortant ou rentrant.

Je ne vois pas ce besoin. Seul l'identifiant de la demande est nécessaire.

#14 - 17 septembre 2019 12:14 - Serghei Mihai

Benjamin Dauvergne a écrit :

Aussi si on vise un seul sftp je pense qu'un seul champ SFTPField suffit, chemin et préfixe peuvent être mis en dur (on parle d'un connecteur contrib/ après tout).

Je préfère garder 2 champs sans hardcoder le chemin. Le connecteur sera utilisé par ailleurs (Orléans est demandeur) et le chemin peut être différent.

En plus, malgré ce que la doc indique:

```
Répertoire des demandes :      publik_vers_gesbac
...
Répertoire des fichiers retours :      gesbac_vers_public
```

aujourd'hui sur le SFTP mis à disposition le répertoire pour charger les fichiers s'appelle "upload". Il faut laisser la main sur les noms des répertoires.

#15 - 17 septembre 2019 16:13 - Frédéric Péters

Le connecteur sera utilisé par ailleurs (Orléans est demandeur)

S'il s'agit d'une vraie application derrière qui peut se retrouver ailleurs et dont l'interfaçage y sera également celui-ci, ça devrait plutôt aller dans passerelle.apps.

#16 - 17 septembre 2019 16:13 - Frédéric Péters

Ah bah ça y est, malgré le "(on parle d'un connecteur contrib/ après tout)" plus haut.

#17 - 17 septembre 2019 16:19 - Serghei Mihai

Frédéric Péters a écrit :

Ah bah ça y est, malgré le "(on parle d'un connecteur contrib/ après tout)" plus haut.

Il s'agit d'un connecteur ré-utilisable et pas spécifique à un client, donc dans apps.
J'ai mis à jour la branche avec prise en compte de tes remarques.

Ta lecture est bonne.

Quand une réponse à une demande est faite, ça vérifie s'il y a déjà un fichier CSV sur le SFTP correspondant à la demande. Ça stocke les données dans un modèle éviter d'interroger le SFTP pour la même demande.

#18 - 19 septembre 2019 08:57 - Benjamin Dauvergne

Je ne suis toujours pas ok avec ce fonctionnement dont je ne vois pas en partie où ça colle avec la doc :

- les appels SFTP ne devrait pas être faits de manière synchrone (sur le GET surtout ça va être affreux) mais en tâche de fond
- tu supposes qu'en ne mettant qu'une demande par fichier .csv et ne le nommant avec l'id de la demande tu auras le retour dans un fichier du même nom, et juste ce retour, je ne vois rien dans le .docx qui nous assure de ça; il est spécifié : plusieurs demandes par CSV entrant, pareil en sortie; ça me paraît bizarre surtout sachant que certaines demandes (d'après la spéc) seront traitées par un agent.

#19 - 19 septembre 2019 09:02 - Benjamin Dauvergne

Serghei Mihai a écrit :

Benjamin Dauvergne a écrit :

Ouais ça va pas du tout, on peut pas gérer un fichier CSV distant comme ça, l'objectif à la lecture du .docx est clairement pas d'avoir du temps réel (ça dit clairement la DSI, tous les soirs, etc...).

J'aurai vu un truc en 2 temps :

- à chaque appel send:
 1. on cherche ou on crée un objet CSV dans un statut "open"
 2. on crée un objet CsvLine avec les données du formulaire, fini (aucun souci de concurrence possible, tout va bien)
- tous les n-période (tous les soirs disont) on prend le fichier CSV open le plus vieux, on le passe en "closed", ensuite on se connecte au FTP on pose un CSV avec l'identifiant incrémental du CSV (genre le .id ça doit suffir), quand c'est terminé on passe l'objet CSV en statut "sent"

Faire cela tous les n-périodes ou au moment de l'appel webservice depuis wcs, je ne vois pas différence.
Que la DSI (ou Gesbac) vienne récupérer les fichiers tous les soirs, toutes les n heures, c'est leur affaire.

Ok, comme tu ne connais pas la fréquence des lecture, tu devrais tout de même ne pas écrire directement dans le fichier et passer par un fichier temporaire puis renommer.

Je ne vois pas ce besoin. Seul l'identifiant de la demande est nécessaire.

C'est perturbant d'avoir appelé ça CSV en fait, même si tu ne produis qu'un fichier CSV par demande ce serait plus clair d'appeler ça Formdata ou un truc du genre.

#20 - 19 septembre 2019 09:57 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Je ne suis toujours pas ok avec ce fonctionnement dont je ne vois pas en partie où ça colle avec la doc :

- les appels SFTP ne devrait pas être faits de manière synchrone (sur le GET surtout ça va être affreux) mais en tâche de fond
- tu supposes qu'en ne mettant qu'une demande par fichier .csv et ne le nommant avec l'id de la demande tu auras le retour dans un fichier du même nom, et juste ce retour, je ne vois rien dans le .docx qui nous assure de ça; il est spécifié : plusieurs demandes par CSV entrant, pareil en sortie; ça me paraît bizarre surtout sachant que certaines demandes (d'après la spéc) seront traitées par un agent.

Et pour le dire plus franchement, la spec renvoie le numéro de la demande dans le CSV je ne vois pas pourquoi on passe par un mécanisme basé sur le nom du fichier alors qu'on a déjà tout ce qu'il faut au niveau du CSV (colonne 2 du CSV, je reconnais que de nous forcer à un entier c'est un peu chiant).

#21 - 19 septembre 2019 10:28 - Serghei Mihai

Benjamin Dauvergne a écrit :

- les appels SFTP ne devrait pas être faits de manière synchrone (sur le GET surtout ça va être affreux) mais en tâche de fond

Dans un deuxième temps j'imaginai un job genre hourly ou daily qui vient interroger le SFTP pour lire les fichiers réponses et stocker tout dans la base.

Comme ça peu d'appels GET déboucheront sur des appels SFTP.

- tu supposes qu'en ne mettant qu'une demande par fichier .csv et ne le nommant avec l'id de la demande tu auras le retour dans un fichier du même nom, et juste ce retour, je ne vois rien dans le .docx qui nous assure de ça; il est spécifié : plusieurs demandes par CSV entrant, pareil en sortie; ça me paraît bizarre surtout sachant que certaines demandes (d'après la spéc) seront traitées par un agent.

Yep, c'est une possibilité, mais nous avons convenu d'envoyer un fichier CSV par demande et recevoir un fichier par retour. Cela facilitera le "debug" à l'éditeur (cette "API" n'a jamais été utilisée nulle part, d'après l'éditeur).

Et ce n'est pas dans la doc non plus, mais nous avons convenu que "xxxxx" dans le suffixe du fichier contiendra la valeur du fichier demande.

Pour revenir à ton idée: l'objet "Formdata" (je renommerai "CSV") stockera les données de l'appel webservice et dans un job daily écrira les fichiers sur le SFTP. Avec les flags qui vont bien pour différencier ceux qui ont été déjà écrits.

#22 - 20 septembre 2019 15:56 - Serghei Mihai

Branche à jour avec possibilité d'écrire un fichier par demande et lecture dans un ou plusieurs fichiers de retour.

#23 - 20 septembre 2019 17:13 - Benjamin Dauvergne

- print row
- des '}' tous seuls sur une ligne mal indentés (merci d'installer vim et khuno)
- moi j'utiliserai bêtement les jobs plutôt que daily pour poser tes formulaires (comme ça c'est fait ASAP mais pas de façon synchrone) et pour la lecture des réponses hourly plutôt.

#24 - 20 septembre 2019 18:29 - Serghei Mihai

- Fichier 0001-gesbac-initial-connector-35325.patch ajouté

Effectivement bonne idée de passer par le jobs pour l'envoi des fichiers.

#25 - 20 septembre 2019 20:56 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Allez go, j'espère que t'as pu tester sur une instance de recette.

#26 - 23 septembre 2019 09:58 - Serghei Mihai

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

Testé en envoyant les fichiers depuis une instance locale vers leur SFTP :

```
commit bf7430176fa5f0e35bb60e92c1386365b0d13407 (HEAD -> master, origin/master, origin/HEAD)
Author: Serghei Mihai <smihai@entrouvert.com>
Date: Fri Sep 6 14:02:28 2019 +0200
```

```
gesbac: initial connector (#35325)
```

#27 - 23 septembre 2019 14:15 - Frédéric Péters

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

Fichiers

interface_Gesbac_Publik_Grenoble_v6.docx	20,7 ko	09 août 2019	Serghei Mihai
0001-gesbac-initial-connector-35325.patch	26,7 ko	17 septembre 2019	Serghei Mihai
0001-gesbac-initial-connector-35325.patch	24,1 ko	20 septembre 2019	Serghei Mihai