Audit et Tests performance

Vous trouverez ci-joint quelques outils pour EzGED à lancer en ligne de commande sur le serveur

tools.zip dézipper ce fichier

Audit de sécurités requêtes

Cet outil permet de générer un export csv des droits par utilisateurs et par requête, dans le répertoire où le script a été lancé.

Exécutez le en ligne de commande avec la syntaxe suivante

```
python sec_dump.py instance 1 1
```

Audit de performances des requêtes publiées

Permet de réaliser un audit de performance de vos requêtes et de vous indiquer une éventuelle requête trop longue, indiquant souvent une mauvaise optimisation de la requête (problème de jointures et/ou de clé).

Exécutez le en ligne de commande avec la syntaxe suivante

```
python keepalive_tester.py instance 1 1
```

Astuce : tester également les requêtes non publiées : mettez en commentaire les lignes suivantes dans le script (en ajoutant un #):

```
#pub = ["@fld", "_qry", "QRY_PUBCNTCOR"], ["@op", "="], ["@val", 1]
#tblquery.cond = pub
```

Reprise Chorus Pro 18/02/2022

La mise-à-jour du Portail Chorus Pro en v3.0.1 (voir Portail-services-Chorus-PRO-changements-principaux.pdf) a conduit à des problèmes d'intégration des factures récupérées sur le portail entre le 18/02/2022 et le 21/02/2022 vers 18h.

Nature du problème: le format du fichier pivot. Au lieu de transmettre le fichier pivot au format xml, la plateforme transmettait des fichiers zip contenant le fichier pivot xml ainsi que les fichiers joints (la facture pdf, etc).

Afin de faciliter le traitement de ces fichiers et de ne pas avoir à procéder à une nouvelle récupération de ces factures, nous proposons des scripts de reprise selon les cas suivants:

Cas 1: la facture au format zip est archivée

Dans ce cas, l'intégration de la facture au format zip dans EzGED n'a pas provoqué d'erreurs au cours du travail de COLD. La fiche a été créée et le fichier archivé est rattaché à celle-ci.

Le fichier a pour nom **Facture_Pivot.xml** mais il s'agit d'un fichier zip.

Pour traiter ce cas veuillez utiliser le script chorus_zip_extract

Cas 2: la facture est bloquée dans le cold

Dans ce cas l'intégration n'est pas réalisée. Il n'y a pas encore de fiche descriptive. Le répertoire du travail contient un fichier **Facture_Pivot.xml** qui est en fait un zip contenant le vrai fichier pivot au format xml ainsi que les pièces jointes (facture pdf, etc).

On va procéder en deux temps:

1. Exécuter le script chorus_cold_fix qui va extraire les données du vrai fichier pivot xml dans la maquette et relancer l'analyse cold. La facture va donc s'archiver au format zip, mais on aura toutes les données nécessaires pour l'analyse.
2. Exécuter le script chorus_zip_extract qui va extraire le fichier pivot et la facture du zip déjà archivé.

Cas 3: fichiers extraits à rattacher à une pochette de workflow existante

Dans le cas où votre cold qui intègre les factures Chorus prévoi un envoi automatique dans le workflow, vous êtes alors dans la situation où le fichier pivot au format zip a été envoyé mais pas les fichiers qu'il contient puisqu'ils auront été extraits après coup.

Mais vous souhaitez que ces fichiers, après extraction, soient rattachés à la même pochette de workflow.

Pour cela vous allez exécuter une deuxième fois (la première servant à traiter le cas 2 si besoin) le script chorus_cold_fix.

Scripts de reprise

Les scripts sont à télécharger ici :

chorus_fix_scripts.zip

chorus_zip_extract

Voir scripts_de_reprise pour obtenir le script.

Ce script recherche les fichier pivot au format zip pour en extraire le contenu et le stocker sur la même fiche descriptive.

Il va dans un deuxième temps identifier les fichiers Facture_Pivot.zip qui sont dans le workflow afin de rattacher à la même pochette les fichiers extraits du zip. Il peut être nécessaire de lancer le script une deuxième fois pour ça, en prenant soin d'attendre que l'archivage des fichiers extraits soient complètement terminé (surveiller les travaux en cours).

Ce que fait le script:

1. Extraire les fichiers du fichier Facture_Pivot.xml qui en fait un zip
    1. Liste numérotée
    2. Cherche les fichiers nommés *Facture_Pivot.xml* depuis une date donnée en entrée.
    3. Pour chaque fichier trouvé
        1. Si le fichier n'est pas un zip, passe au suivant.
        2. Si le fichier zip est vide, passe au suivant.
        3. Créé un travail de stockage (sans l'activer)
        4. Extrait tout les fichiers du zip.
        5. Copie les fichiers extraits dans le répertoire du travail de stockage.
        6. Ecrit le fichier files.lst (liste des fichiers à stocker) dans le répertoire du travail de stockage.
        7. Active le travail de stockage.
        8. Modifie le nom du fichier zip de *Facture_Pivot.xml* à *Facture_Pivot.zip* (et corrige son type mime enregistré dans _fsfile)
2. Attacher à la même pochette workflow que le fichier Facture_Pivot.zip les fichiers qui en ont été extraits

Paramètres

Nom du paramètre	**Description**	**Exemple**
instance	Instance EzGED	instance
jobqueueid	ID du job	1
secusrid	ID utilisateur	1
sincedate	Date au format yyyy-mm-dd hh:mm:ss ou yyyy-mm-dd	2022-02-18

Exemple d'appel

```
>python chorus_zip_extract.py instance 1 1 "2022-02-18"
```

Exemple de sortie pour un fichier à traiter:

```
Looking for Facture_Pivot zip files since 2022-02-18 00:00:00

        Found file #19769 (Table: one, RSID: 553)
        Is a zip file. Process it.
        Storage job created (#115332)
                Extract PJ00FAC77605646700587005930496 0.pdf to
D:\nchp\var\spool\ezged\instance\jobd\1\19769\PJ00FAC77605646700587005930496
0.pdf
```

```
                Copy
D:\nchp\var\spool\ezged\instance\jobd\1\19769\PJ00FAC7760564670058700593049
0.pdf to
D:\nchp\var\spool\ezged\instance\jobd\115332\PJ00FAC776056467005870059304960
.pdf
                Extract PivotS.xml to
D:\nchp\var\spool\ezged\instance\jobd\1\19769\PivotS.xml
                Copy
D:\nchp\var\spool\ezged\instance\jobd\1\19769\PivotS.xml to
D:\nchp\var\spool\ezged\instance\jobd\115332\PivotS.xml
        Writing files.lst
        Activate job #115332
        Rename file #19769 from Facture_Pivot.xml to Facture_Pivot.zip
```

chorus_cold_fix

Voir scripts_de_reprise pour obtenir le script.

Le but du script est d'extraire le contenu du fichier pivot au format xml contenu dans le zip et de l'ajouter à la maquette. Ce script est donc à utiliser uniquement dans le cas où l'intégration des factures est bloquée dans le COLD (à l'étape d'analyse ou d'insertion).

Ce que fait le script:

1. Cherche les travaux en erreur pour un type de COLD donné et une étape donnée (facultatif).
2. Pour chaque travail trouvé:
    1. Si déjà traité, passe au suivant.
    2. Si fichier Facture_Pivot.xml non trouvé, passe au suivant.
    3. Si fichier Facture_Pivot.xml n'est pas un zip, passe au suivant.
    4. Parcours la liste des fichiers du zip:
        1. Si le fichier PivotS.xml est trouvé, ajoute son contenu au fichier maquette (metadata.txt)

Paramètres:

Nom du paramètre	Description	Exemple
instance	Instance EzGED	instance
jobqueueid	ID du job	1
secusrid	ID utilisateur	1
jobtplid	ID du scénario cold	72
changetostep	Numéro de l'étape d'analyse	5
filteronstep	Numéro de l'étape des jobs à traiter (optionnel)	6

Exemple d'appel sans filtre sur une étape

```
>python chorus_cold_fix.py instance 1 1 259 5
```

Exemple d'appel en filtrant sur l'étape 6

```
>python chorus_cold_fix.py instance 1 1 259 5 6
```

Exemple de sortie:

```
Looking for jobs in error where jobtplid = 259 and step =

        Checking job #115331
        File Facture_Pivot.xml is a zip file. Process it.
        Append PivotS.xml content to metadata.txt
        Update job #115331 to step 5 and activate it.
        Flag job #115331 as done.
```

From:
http://wiki.ezdev.fr/ - **EzGED Wiki**

Permanent link:
http://wiki.ezdev.fr/doku.php?id=tools

Last update: **2023/03/17 09:56**