**Scorebot**

Documentation de la conversion en RDF

Table des matières

[Table des matières 1](#_Toc1)

[Prérequis 2](#_Toc2)

[Organisation des fichiers et répertoire 2](#_Toc3)

[Structure de répertoires 2](#_Toc4)

[Structure du répertoire « input » 2](#_Toc5)

[XML à convertir 2](#_Toc6)

[Vocabulaires contrôlés 3](#_Toc7)

[Structure du répertoire « work » 3](#_Toc8)

[Principe de la conversion 3](#_Toc9)

[Convertir les vocabulaires contrôlés 4](#_Toc10)

[Copier les fichiers 4](#_Toc11)

[Convertir les fichiers 4](#_Toc12)

[Valider les données 5](#_Toc13)

[Chargement des données dans GraphDB 5](#_Toc14)

[Annexe : script Python d’analyse des logs 8](#_Toc15)

[Annexe : correction manuelle des problèmes d’encodage des fichiers d’Aloes 11](#_Toc16)

# Prérequis

1. Avoir [Java](https://www.java.com/fr/) installé sur le poste de travail, et avoir la commande « java » dans le PATH

# Organisation des fichiers et répertoire

## Structure de répertoires

**Dossiers**:

* input : contient les données source pour la conversion
* xslt : contient les XSLT de conversion
* doc: contient la doc
* shacl : description du modèle en SHACL permettant une validation des données

Deux autres dossiers seront créés lors de la conversion :

* output : contient le résultat de la conversion
* work : contient les fichiers temporaires, dont les logs, générés à chaque lancement de la conversion

Un autre dossier sera créé lors de la validation :

* output-validation : contient le rapport de validation ainsi que les logs du processus de validation

**Fichiers**:

Le répertoires contient un certain nombre de fichiers, en particuliers les scripts de conversion et de validation : **scorebot\_conversion.bat** et **scorebot\_validation.bat**

## Structure du répertoire « input »

Ce dossier est le dossier principal où se trouvent les fichiers xml à convertir.

Il contient 2 grandes familles de fichiers : les XML à convertir, et les vocabulaires contrôlés utilisés lors de la conversion, en RDF (n’importe quelle sérialisation, typiquement Turtle ou RDF/XML).

### XML à convertir

Les répertoires contenant les XML à convertir sont les suivants :

* input\collectivites : doit contenir le fichier XML d’export des autorités collectivités
* input\partitions : doit contenir le fichier XML d’export des partitions
* input\personne : doit contenir le fichier XML d’export des autorités personnes
* input\thesaurus : doit contenir le fichier XML d’export du thesaurus

Chaque répertoire peut contenir 1 ou plusieurs fichiers XML. S’il y en a plusieurs, chaque fichier XML sera transformé par la XSLT correspondante. Le nom des fichiers n’a pas d’importance : chaque fichier d’input donnera lien à un fichier du même nom dans le répertoire d’output.

### Vocabulaires contrôlés

Les vocabulaires contrôlés sont organisés dans les répertoires suivants :

* input\vocabulaires: contient des sous-répertoires pour chaque vocabulaire contrôlé utilisé lors de la conversion des données :
  + input\vocabulaires\iaml : vocabulaire IAML
  + input\vocabulaires\mimo : vocabulaire MIMO
  + input\vocabulaires\lexvo : vocabulaire Lexvo (langues)
  + input\vocabulaires\responsabilite : vocabulaires des responsabilités (responsabilité des musiciens)
  + input\vocabulaires\role : vocabulaire des rôles / fonctions (auteur, traducteur, etc.)
  + input\vocabulaires\educational-level : vocabulaire des niveau de compétence (premier cycle, etc.)

Chaque répertoire peut contenir 1 ou plusieurs fichiers RDF dans n’importe quelle sérialisation RDF (RDF/XML ou Turtle, etc.). Le nom des fichiers n’a pas d’importance . Le contenu de chaque répertoire de vocabulaire contrôlé est fusionné et normalisé en un seul fichier RDF/XML avant la conversion.

## Structure du répertoire « work »

Lors du lancement du script **scorebot\_conversion.bat**, les dossiers suivants sont créés :

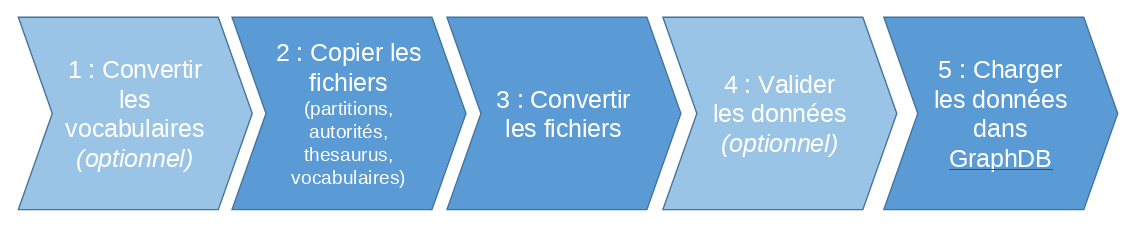
* work\logs : logs de la conversion des fichiers.
* work\vocabulaires\_rdf-xml : répertoire temporaires contenant les versions normalisées de tous les vocabulaires contrôlés.

Le répertoire « work » peut être supprimé sans risque.

Un utilitaire optionnel décrit en annexe permet de faire un parsing du fichier de log.

# Principe de la conversion

Le principe général de conversion est le suivant :



# Convertir les vocabulaires contrôlés

Cette étape est optionnelle : si les vocabulaires contrôlés n’ont pas été modifiés, il n’est pas nécessaire de les convertir.

Cette étape est réalisée manuellement.

Pour convertir un vocabulaire contrôlé :

1. Aller à l’adresse du convertisseur <https://skos-play.sparna.fr/play/convert>.
2. Charger un fichier Excel de vocabulaire dans le champ “dans un fichier sur mon ordinateur”.
3. Laisser toutes les autres options par défaut et cliquer sur le bouton “Convertir” en bas du formulaire.
4. Sauvegarder le fichier Turtle résultat de la conversion, en le sauvegardant dans un des répertoires d’input des vocabulaires contrôlés.
5. Répéter l’opération pour les autres fichiers.

# Copier les fichiers

Copier les fichiers dans les répertoires d’input documentés ci-dessus. Chaque répertoire doit contenir (au moins) un fichier. Si certaines données n’ont pas été modifiées depuis la dernière conversion, il n’est pas utile de les mettre à jour.

L’ensemble des fichiers est reconverti systématiquement, il n’est pas possible de convertir seulement un fichier.

# Convertir les fichiers

1. Ouvrir une fenêtre de “ligne de commandes” dans le répertoire du convertisseur.
2. Lancer **“scorebot\_conversion.bat”.** Le traitement peut prendre entre environ 30 minutes et 1 heure, en fonction de la puissance de la machine.[[1]](#footnote-2)
3. Vérifier que le script a généré :
   1. Des fichiers RDF résultat de la conversion dans le répertoire output.
   2. Des fichiers de log dans le répertoire work\logs
4. Vérifier les fichiers logs dans le repertoir work\logs pour vérifier s’il y a des erreurs.

# Valider les données

Cette étape est optionnelle.

1. Ouvrir une fenêtre de “ligne de commandes” dans le répertoire du convertisseur.
2. Lancer **“scorebot\_validation.bat”**
3. Vérifier que le script a généré le rapport de validation dans “output-validation\scorebot\_validation-report.html”
4. Ouvrir le fichier “output-validation\scorebot\_validation-report.html” pour consulter le rapport de validation.

# Chargement des données dans GraphDB

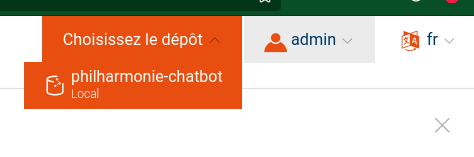
Une fois la conversion réalisée on doit charger les fichiers de données sur GraphDB.

Aller à GraphDB à l’adresse suivante :

* <https://graphdb.prd.iumio.fr/login>
* Se connecter avec la compte d’Administrateur :
  + Utilisateur : **admin**
  + Mot de passe : **pGraphDB\_**

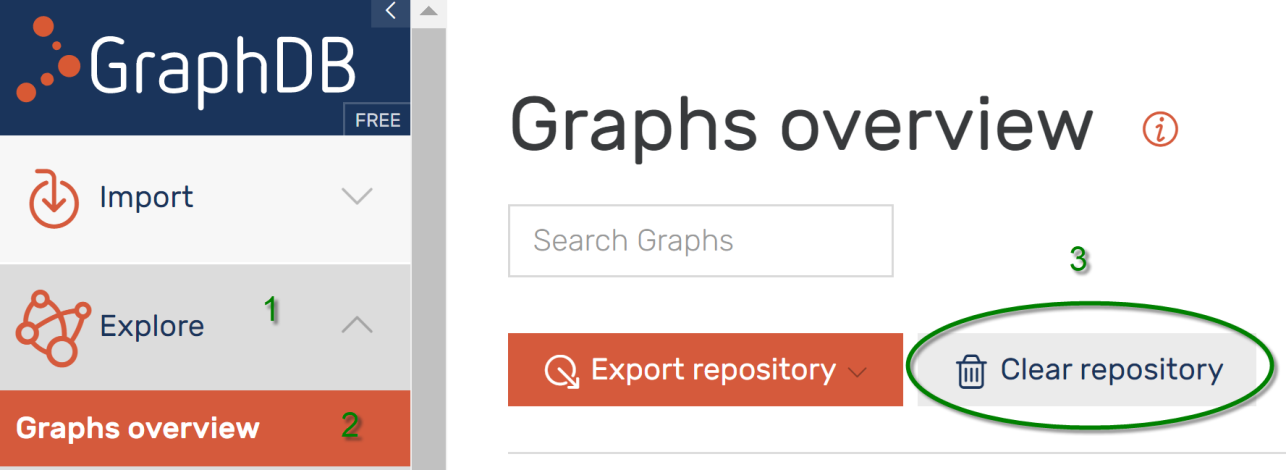
Une fois connectés, suivre les étapes suivantes :

Se connecter au repository « philharmonie-chatbot » par le menu en haut à droite :

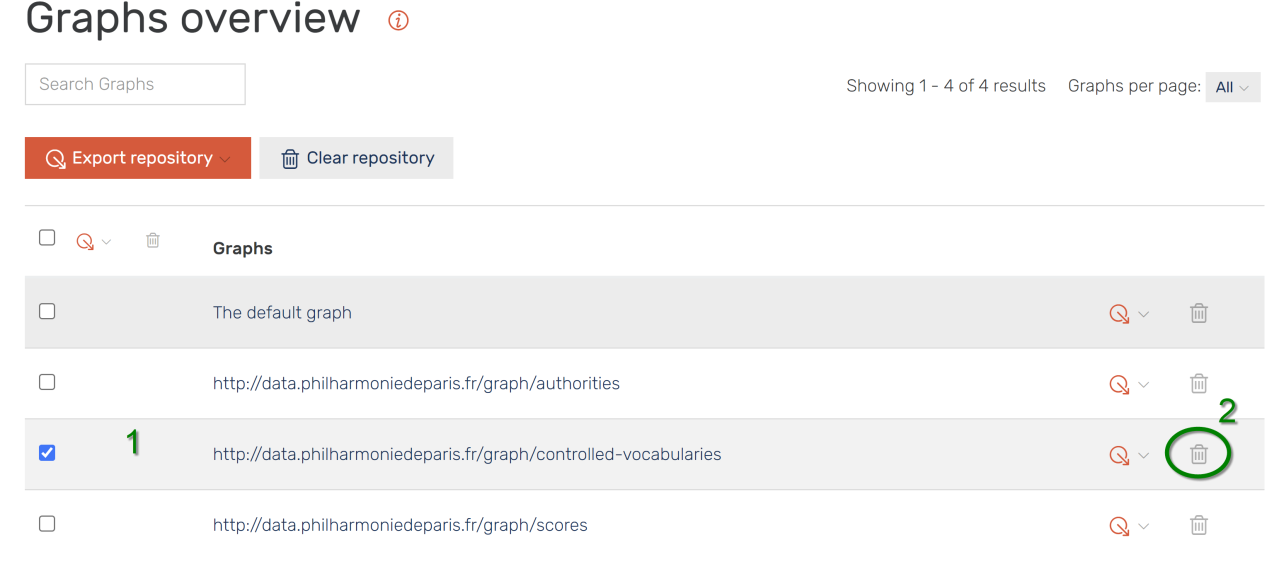


Si le repository a déjà des données, il faut les supprimer avant d’uploader les nouvelles données fichiers :

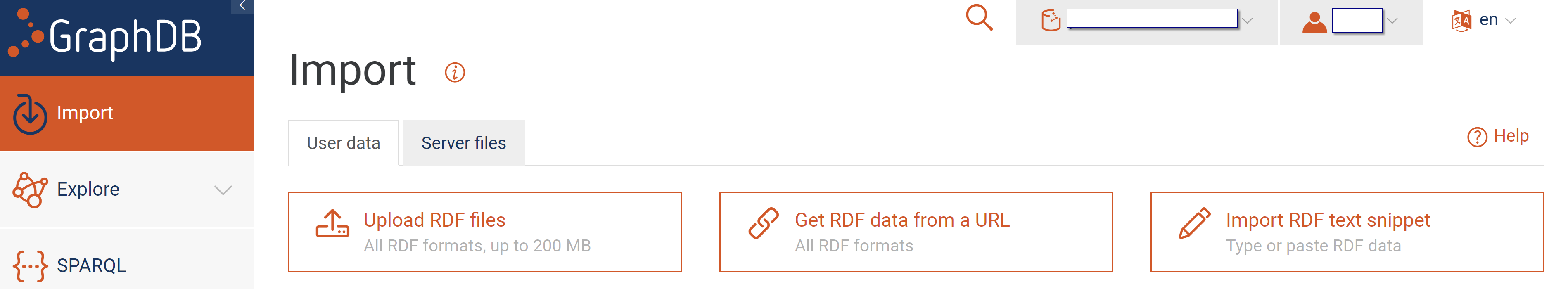
* Dans le menu de gauche, aller à Explorer > Graphs overview :
  + Si vous souhaitez supprimer *toutes les données*, Cliquez le bouton Clear repository. Attention cela vide complètement la base ! Toutes les données devront ensuite être rechargées.



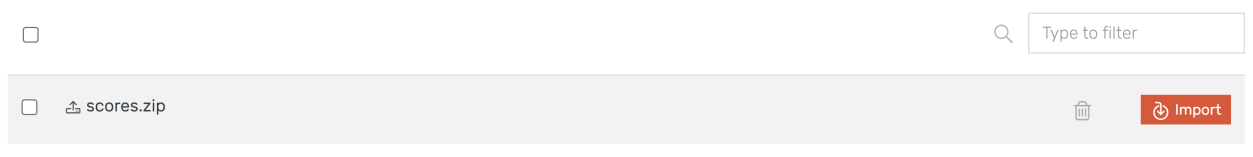
* + Si vous voulez supprimer une partie des données seulement pour les recharger (par exemple seulement les partitions, ou seulement les vocabulaires contrôlés, ou seulement les autorités), vous devez choisir le graph (1) es cliquez sur le bouton Supprimer (2).



* + Confirmer l’action a demandé avec le bouton Yes.
* Uploader les données
  + Il faut zipper les données pour les charger, en créant 3 zips :
    - 1 fichier zip contenant l’output RDF de la conversion des partitions
    - 1 fichier zip contenant l’output RDF de la conversion des fichiers d’autorités et du thesaurus :
      * Autorités collectivités
      * Autorités personnes
      * Thesaurus
  + 1 fichier zip contenant l’output RDF de tous les vocabulaires contrôlés. Attention les vocabulaires contrôlés sont à récupérer dans le répertoire work\vocabulaires\_rdf-xml.
  + Dans le menu de gauche, naviguer dans Import



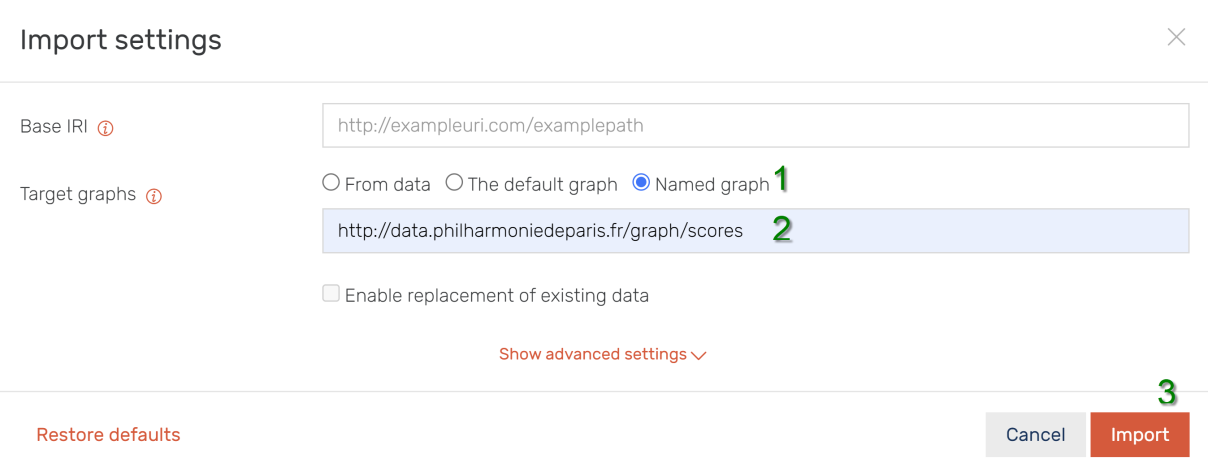
* Cliquez sur l’option Upload RDF files. Choisissez le fichier zip des Partitions et chargez-le.



* + Sélectionner la case à gauche du nom du fichier chargé et cliquez sur le bouton Import (2)



* Dans la boite de dialogue qui s’affiche, choisissez l’option Named graph(1), et préciser l’URI du graphe nommé (2), dans cette cas sera <http://data.philharmoniedeparis.fr/graph/scores> et pour terminer cliquez Import(3).

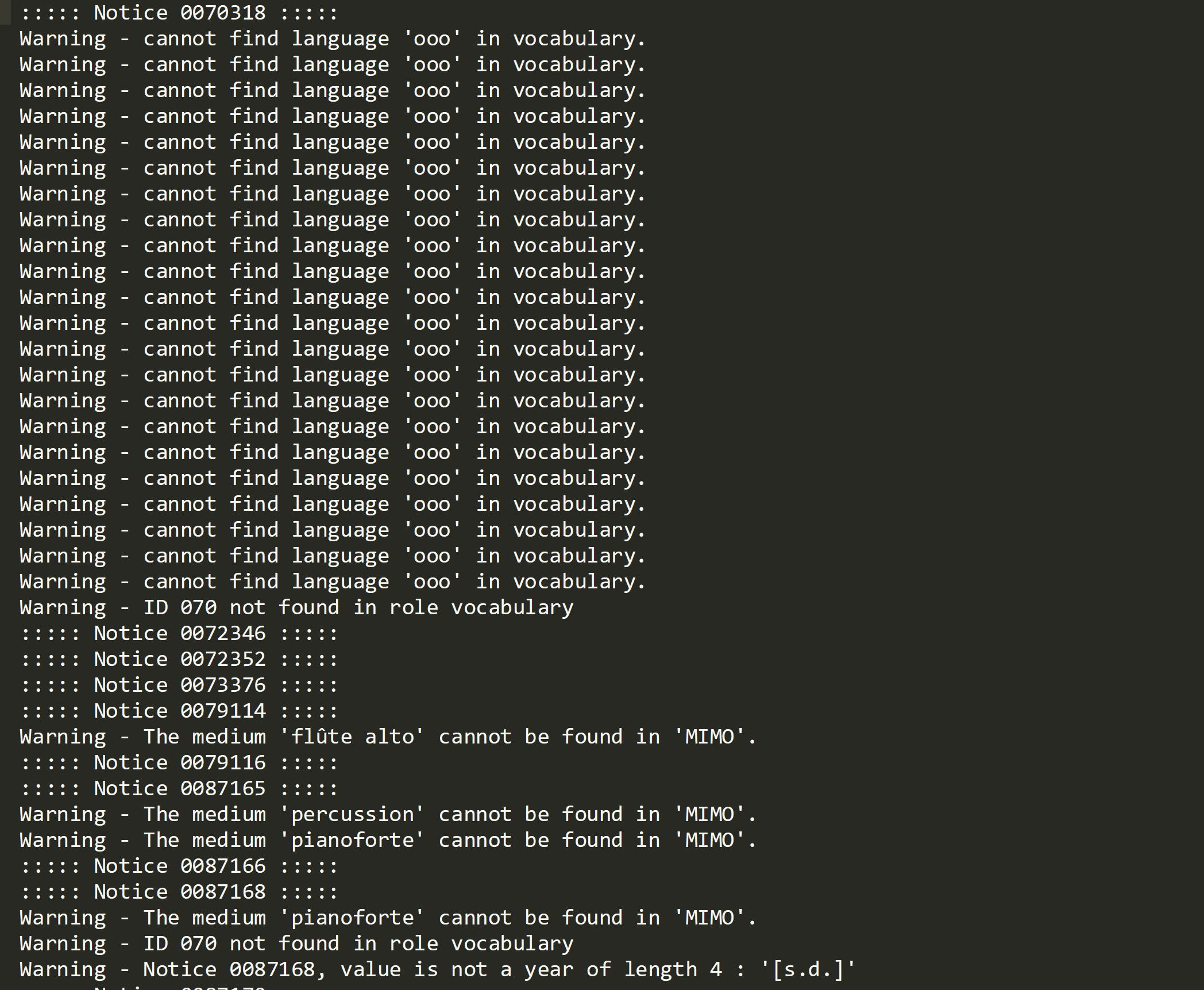


* + Répétez la même procédure pour le fichier d’Autorités avec l’URI [du graphe nommé http://data.philharmoniedeparis.fr/graph/authorities](http://data.philharmoniedeparis.fr/graph/authorities)
  + Répétez la même procédure pour le fichier des vocabulaires contrôlés avec l’URI du graphe nommé <http://data.philharmoniedeparis.fr/graph/controlled-vocabularies>

# Annexe : script Python d’analyse des logs

L’utilitaire rapport\_log\_philhar.py est un script python qui permet de fournir un rapport avec tous les messages des résultats qui existe dans un fichier log.

Exemple de message:



**Prérequis :**

Python doit être installé :<https://www.python.org/downloads/>

**Comment utiliser le script.**

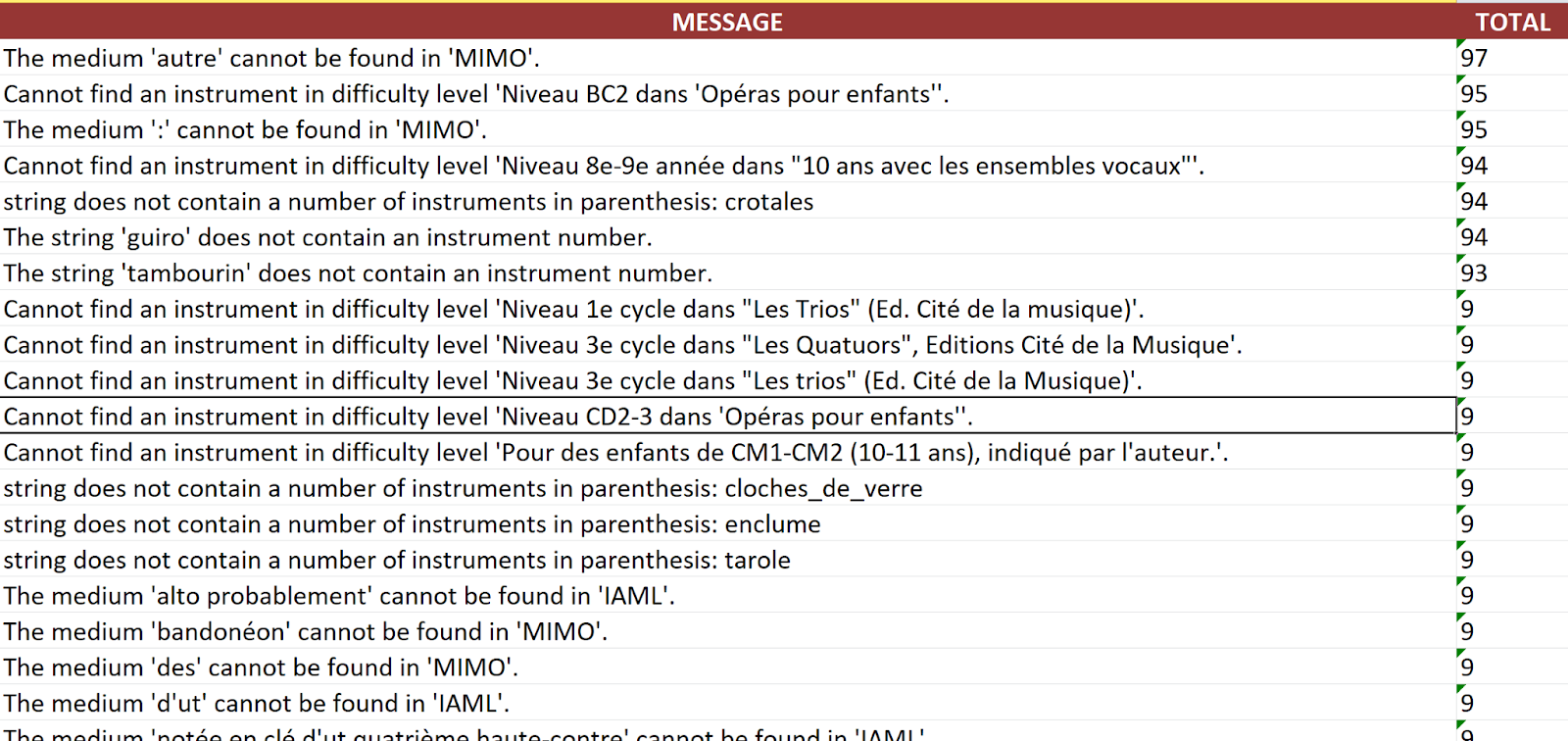
L’outil a besoin des inputs suivant:

* Chemin du fichier de log à analyser (*Obligatoire*) : un fichier de log du convertisseur dans le répertoire work\logs
* Chemin du fichier de sortie (*Obligatoire*): avec l’extension \*.csv
* Mot-clé à chercher (Optionel) :
* Optional: Si vous avez besoin de chercher un mot clé, vous pouvez l’ajouter sans problème.

Par exemple :

python rapport\_log\_philhar.py C:\Temp\philharmonie\work\logs\ExportPartitions-20211201.log C:\Temp\philharmonie\scorebot\_analyse.csv

Le résultat sera un tableau avec les messages de Warning trouvés dans le fichier, et le nombre de fois où le message est apparu :



Exemple d’exécution avec un filtre de mot clé:

python rapport\_log\_philhar.py C:\Temp\philharmonie\work\logs\ExportPartitions-20211201.log C:\Temp\philharmonie\scorebot\_analyse.csv "Les quatuors"

Seules les lignes contenant ce mot-clé seront sorties :



# Annexe : correction manuelle des problèmes d’encodage des fichiers d’Aloes

Le script de conversion a besoin de fichiers XML valides en entrée. Si les fichiers XML contiennent des problèmes d'encodage ou des caractères invalides, il n'est pas possible d'appliquer des feuilles de style XSLT dessus.

Il est arrivé que les fichiers d'export de partitions et/ou les fichiers d'export de personnes contiennent des caractères invalides. La seule solution est de corriger ces problèmes manuellement; voici la procédure que nous avons suivi:

Prérequis : cette procédure s'exécute avec une série de commandes sous Linux.

**Etape 1 : caractériser le problème**

Pour vérifier la conformité XML d'un fichier, on utilise l'utilitaire xmllint pour demander un formattage (« pretty print ») du fichier avec l’option « --format ». Si la commande échoue le fichier n'est pas conforme:

xmllint --format ExportPartitions-20211201.xml > ExportPartitions-20211201\_indented.xml

on peut obtenir des erreurs comme:

ExportPartitions-20211201.xml:3: parser error : Opening and ending tag mismatch: Zwei line 3 and data

CHAMP><SOUSCHAMP UnimarcSubfield="462$a"><data><Zwei > Chinesische Lieder</data>

^

ExportPartitions-20211201.xml:3: parser error : Opening and ending tag mismatch: data line 3 and SOUSCHAMP

HAMP UnimarcSubfield="462$a"><data><Zwei > Chinesische Lieder</data></SOUSCHAMP>

^

ExportPartitions-20211201.xml:3: parser error : Opening and ending tag mismatch: SOUSCHAMP line 3 and champs

USCHAMP UnimarcSubfield="462$f"><data>Viktor Ullmann</data></SOUSCHAMP></champs>

^

**Etape 2 : insérer des sauts de ligne dans le fichier source**

Un problème est que le fichier de départ est écrit sur une seule ligne, et que les messages d'erreur indiquent tous la ligne "3", ce qui rend l'identification et la correction de l'erreur impossible. Pour cela on va ajouter un saut de ligne à la fin du XML de chaque notice avec la commande sed :

sed 's/<\/NOTICE>/<\/NOTICE>\n/g' ExportPartitions-20211201.xml > ExportPartitions-20211201-newlines.xml

Maintenant la même commande xmllint que précédemment va indiquer des numéros de ligne :

xmllint --format ExportPartitions-20211201-newlines.xml > ExportPartitions-20211201-newlines\_indented.xml

ExportPartitions-20211201-newlines.xml:34715: parser error : Opening and ending tag mismatch: Zwei line 34715 and data

CHAMP><SOUSCHAMP UnimarcSubfield="462$a"><data><Zwei > Chinesische Lieder</data>

^

ExportPartitions-20211201-newlines.xml:34715: parser error : Opening and ending tag mismatch: data line 34715 and SOUSCHAMP

HAMP UnimarcSubfield="462$a"><data><Zwei > Chinesische Lieder</data></SOUSCHAMP>

(...)

On a bien le numéro de ligne "34715" dans le message d'erreur.

**Etape 3 : corriger le fichier**

A partir du numéro de ligne on peut corriger le fichier. La correction peut se faire dans un éditeur texte classique, s'il est capable d'ouvrir un fichier aussi gros. Sinon, avec l'éditeur Linux vi, on peut se positionner sur une ligne donnée en tapant <Echap>, puis ":" suivi du numéro de ligne (par exemple ":34715"). Ou bien en tapant le numéro de ligne, suivi de Shit+g.

Une fois positionné sur la bonne ligne, on peut faire la correction sur le champs problématique, par rapport au message d'erreur précédemment remonté par xmllint.

1. Il est possible de suivre « l’avancement » de la conversion en surveillant la taille des fichiers dans le répertoire d’output. Le fichier d’output global des partitions doit faire environ 224Mo une fois complètement généré. [↑](#footnote-ref-2)