data-merging-documentation

Thomas Bouteille

February 2023

1 Introduction

Ce document explique comment utiliser l'outil de "data-merging". L'objectif de cet outil est de regrouper des données provenant de plusieurs sources différentes selon certains critères (e.g. ne choisir que certaines colonnes, que certaines lignes, etc.). L'idée d'un tel outil vient du fait qu'il s'agît d'une opération couramment effectuée pour différents domaines d'applications. Avoir un tel outil permet donc de rapidement fusionner des sources de données en une seule de type choisit. Si les sources de données sont de grandes tailles ou que l'on souhaite regrouper toutes les sources à un seul endroit commun, l'utilisation d'un ETL est à envisager.

2 Structure

```
data-fetching
├── data-export
│   ├── json
│   └── csv
├── data-sources
│   └── ...
└── script
    ├── readers
    │   ├── reader.py
    │   ├── excel_reader.py
    │   └── walstat_reader.py
    ├── utility
    │   └── utility.py
    ├── main.py
    └── sources.yaml
```

data-export : Contient les données regroupées selon un format en particulier définit par le nom du sous-dossier.

data-sources : Contient tous les fichiers source des données à regrouper.

script : Contient les fichiers du code qui réalise la fusion.

- **reader** : Contient les différents lecteurs de type de données possible. Actuellement, il est possible de merger des données provenant des sources xls et de l'api de walstat.
- **utility** : Contient les méthodes utilitaires du script.
- **main.py** : Le point d'entrée du script
- **sources.yaml** : Le fichier de configuration à modifier pour indiquer les sources et leurs options pour la fusion.

L'implémentation des "readers" ce fait via un design "Template". Cela permet d'avoir un template pour rajouter d'autre lecteurs si nécessaire. Ci-dessous le schéma de classe du "reader".

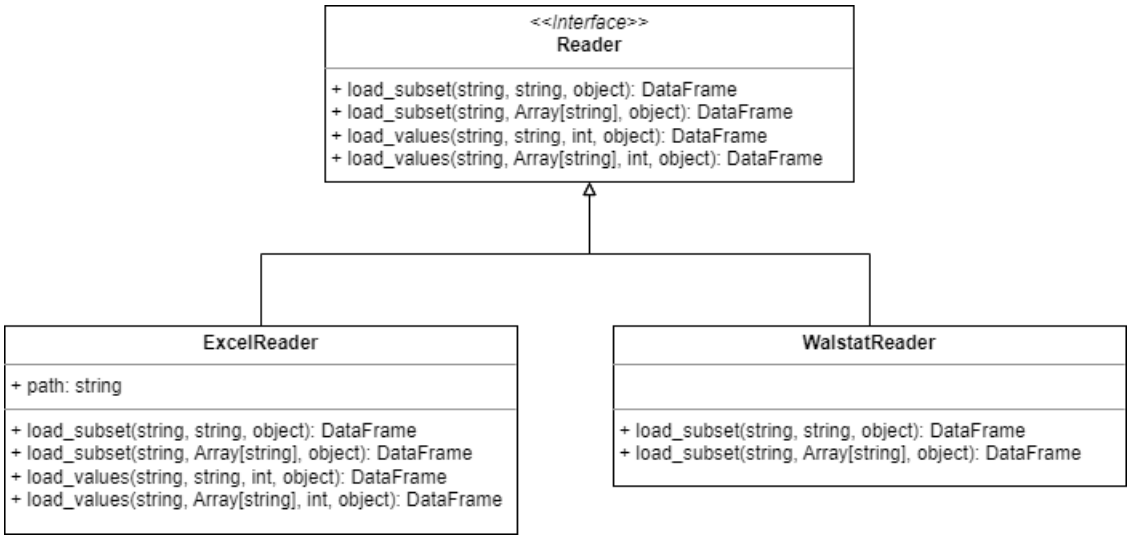

Figure 1: Schéma de classe du lecteur de fichier

3 Documentation de sources.yaml

Plusieurs scénarios sont possibles actuellement. Soit la source de données est un fichier de type excel, soit il s'agit de l'api de walstat. De plus, il est possible

de récupérer soit, un sous-ensemble de plusieurs données de la source, soit une seule valeur.

Le type de fichier *yaml* a été choisi car il est facilement interpréter par *Python* et qu'il possède une notion de typage des données. L'utilisation du type *json* aurait pû être envisager également. L'avantage principal qu'offre *yaml* par rapport à *json* est l'inclusion possible de commentaire directement dans le fichier. Les deux types de fichier se concordant bien l'un l'autre, on peut imager que le changement d'un type à l'autre ne devrait pas posé de problème. Cependant, cela nécessitera de modifier un peu le code pour ouvrir un fichier *json* au lieu de *yaml*.

3.1 Structure commune d'une entrée

Arguments	Type	Description
type	string	Le type de récupération que l'on souhaite effectuer. Peut prendre comme valeur : • **"frame"** Récupération d'un sous-ensemble de valeurs dans la source. • **"cells"** Récupération d'une seule valeur dans la source.
source_type	string	Le type de source des données. Peut prendre comme valeur : • **"xls"** Récupération d'un fichier de type excel. • **"walstat"** Récupération depuis l'api walstat.
look_in	Array	Un tableau de clé-valeur où la clé est un *string* correspondant à la feuille à utiliser et la valeur est un *string* correspondant aux colonnes à utiliser séparé par une virgule.
[outer_group_on]	string	Le nom de colonnes à utiliser pour merger avec les sources précédentes.
[inner_group_on]	string	Le nom de la colonne à utiliser pour merger entre les différentes feuilles utilisées de la source de données.

3.2 Import d'un sous-ensemble excel

Arguments	Type	Description
path	string	Le chemin d'accès du fichier. Chemin absolu ou relatif au fichier *main.py*
options	Object	Un object regroupant les options à utiliser lors du regroupement. • **skiprows** Un tableau correspondant à la plage d'index des lignes à ne pas considérer dans le fichier. (Généralement utilisé pour enlever les ligne vide au début d'un fichier.). Ou alors un entier correspondant à l'index d'une ligne à ne pas considérer. • **columns_rename** Un tableau contenant les nouveaux noms des colonnes. Ce tableau doit avoir la même taille que le nombre de colonnes considéré dans look_in. Ou alors un objet qui contient pour une feuille définie dans look_in un ensemble de clé-valeur où la clé correspond à l'ancien nom de la colonne et la valeur au nouveau nom de la colonne.

3.3 Import d'une valeur excel

Arguments	Type	Description
path	string	Le chemin d'accès du fichier. Chemin en absolu ou relatif au fichier *main.py*.
rows	Array	Tableau d'index de ligne à récupérer.
options	Object	Un object regroupant les options à utiliser lors du regroupement. • **columns_rename** Un tableau contenant les nouveaux noms des colonnes. Ce tableau doit avoir la même taille que le nombre de colonnes considéré dans look_in. Ou alors un objet qui contient pour une feuille définie dans look_in un ensemble de clé-valeur où la clé correspond à l'ancien nom de la colonne et la valeur au nouveau nom de la colonnes. • **transpose** Booléan qui indique si la source de données doit être transposée. • **title_row_index** Index de la ligne à considéré comme nom de colonne. Cette option ne doit être utilisé qu'avec une récupération de type "cells".

3.4 Import d'un sous-ensemble WalStat

Arguments	Type	Description
options	Object	Un object regroupant les options à utiliser lors du regroupement. • **period** Un ensemble de clé-valeurs où la clé correspond à un composant de look_in et la valeur est un tableau d'années à considérer. Si aucune période pour un composant de look_in n'est donnée, alors la valeur pour la dernière année encodée est récupérée. • **columns_name** Un ensemble de clé-valeurs où la clé correspond à un composant de look_in et la valeur est un tableau de nom de colonnes à attribuer pour ce composant. En sachant qu'aucun nom de colonnes n'est donné par l'api walstat, donc si aucune valeur n'est donnée pour un composant la colonne aura simplement le nom *"valeur"*.