UNIVERSITÉ RENNES 2

Projet POO

Xavier André & Romain Tavenard

1 Préambule

Pour ce projet, vous devrez travailler par groupes de 2 ou 3 et votre rendu se fera sous la forme d'un fichier Python. Ce fichier devra être nommé `P03.py` et contenir, sous la forme de commentaires dans l'en-tête du fichier :

- les noms et numéros étudiant de tous les membres du groupe ;
- en cas d'usage de ChatGPT ou d'un autre assistant, un lien vers la session utilisée pour s'aider (lien "Share Chat" en haut à droite de la fenêtre ChatGPT)

comme dans l'exemple suivant :

```
# 22000002 Paul Machin
# 22000227 Yolène Truc
# ChatGPT https://chat.openai.com/share/04760567-235e-40a4-b60d-330877b57e

# Section 1 : les imports
# [...]
```

La date limite de rendu est indiquée sur CURSUS dans l'espace de dépôt.

2 Sujet

Dans ce projet, vous allez implémenter des classes permettant de lire des fichiers de données et de stocker leur contenu sous la forme de listes de dictionnaires.

Votre code devra permettre de lire des fichiers aux formats JSON, CSV et XLS(X) [1].

Pour chaque type de fichier, une classe devra être implémentée pour laquelle les méthodes suivantes devront être disponibles :

1. Vous pourrez utiliser le package `pyexcel` pour cela

- `__init__` : prend en entrée un nom de fichier, stocké dans l'attribut public `fname`, et charge le contenu de ce fichier dans un attribut privé `_data` de type liste de dictionnaires ;
- `_parse` : lit le fichier pointé par l'attribut `fname` et stocke son contenu dans `_data`
- `_update_keys` : stocke dans l'attribut public `keys` la liste des clés présentes dans au moins un dictionnaire de la liste `_data`, triées dans l'ordre alphabétique ;
- `__getitem__` : permet d'accéder à des tranches de données issues de `_data` ;
- `__repr__` : devra donner une sortie du type

  ```
  <JSONReader('truc.json'), 2 rows, keys: ['a', 'b', 'c']>
  ```

- `reload` : recharge le contenu du fichier pointé par l'attribut `fname` dans `_data`
- `from_attributes` : prend en entrée une liste `l_attr` et retourne une liste des données de `_data` restreintes aux attributs listés dans `l_attr`.

Vous utiliserez toutes les possibilités offertes par la programmation orientée objet pour limiter la répétition de code.

Vous pourrez tester votre code sur les fichiers `chose.xlsx`, `machin.csv` et `truc.json`.

2.1 Bonus

Si vous avez déjà implémenté avec succès ce qui est demandé ci-dessus, vous pourrez vous pencher sur le bonus suivant.

Proposez un code qui permet d'obtenir les sorties suivantes :

```
>>> c = ReaderCollection(["truc.json", "machin.csv", "chose.xlsx"])
>>> print(c)
[<GenericReader('truc.json'), 2 rows, keys: ['a', 'b', 'c']>,
 <GenericReader('machin.csv'), 4 rows, keys: ['a', 'b', 'c', 'd', 'e']>,
 <GenericReader('chose.xlsx'), 3 rows, keys: ['a', 'b', 'c', 'd']>]
>>> from pprint import pprint
>>> pprint(c.from_attributes(["b", "c"]))
[{'b': 45, 'c': None},
 {'b': None, 'c': 76},
 {'b': '43432', 'c': '432432'},
 {'b': '43432', 'c': '432432'},
 {'b': '43432', 'c': '432432'},
 {'b': '43432', 'c': '432432'},
 {'b': 2, 'c': 3},
 {'b': 6, 'c': 7},
 {'b': 10, 'c': 11}]
```