trodicaro / **buckets**

Sample ETL project with tests. Edit

#etl #python-3-6 #csv #unit-test #json-parser #file-writing #object-oriented-programming #dictionaries Manage topics

57 commits | **1** branch | **0** releases | **1** contributor

Branch: **master** | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download**

trodicaro Update README.md — Latest commit `e292909` now

README.md	Update README.md	now
bucket_collection.py	Finish test of dup keys to be empty	2 years ago
bucket_collection_test.py	Edits for running the original data provided instead of my testing sa…	2 years ago
purchase_buckets.csv	Add the source files	2 years ago
purchase_data.csv	Add the source files	2 years ago

README.md

Buckets

Given a csv file with "buckets" and another csv file with purchase records, categorize the records into buckets based on some specificity rules.

Assumptions Assumed English words in UTF-8 in the files, thus didn't use any character decoding (https://stackoverflow.com/questions/6797984/how-to-convert-string-to-lowercase-in-python). Seems like Python3 handles it though.

If the "*,*,*" bucket does not exist in the purchase buckets, algorithm creates one at the beginning as per the example above.

Design

I initially wanted to use a regex to capture the data fields needed from the csv file, but it became too cryptic to read and switched to regular string methods. I also learned that reading from a csv can be done with pandas, but I suspect it's an overkill.

My solution started without using classes and object. Looking for the abstractions as I solved.

Testing

Ideally I would run different sets of csv files on every run, but I kept it at one set for now and focused on writing a comprehensive test suite.

Other Notes

Must use python 3.6 or newer. I had a working version with OrderedDict (for lower versions of python, but decided to switch to the regular dictionary structure since ordering is supported starting python3.6).