HelloStreams.py checklist and program information.

Submitted by Barb Dawdy on May 4, 2016

1. A brief document of assumptions that you made interpreting the assignment

Assumptions

* + All storage internal in program was based on store and product (only 1) was not considered at this time.
  + It was assumed that a –n (number) of 200 generated records to the jar file was sufficient for the output needed by the teacher.
  + Since only one product I used a simple dictionary structure which allowed for simple storage of the inventory changes and alerts due to only one product. With more than one product I would need to be more complex data structure which would include storage of the store and product.
  + Frequency was determined by quantity change divided time since last purchase or event. Previous time was stored in a dictionary by store and used with the current event time for that store to get a delta time.
  + Per the class office hrs I assumed it was acceptable to display all information in Q2 from the assignment in the json output event line and also to display the alert message when it occurred.
  + 3 different alert messages were provided: one when inventory =<5 but inventory >0, when inventory = 0, and when inventory goes negative. This could be used for escalation notification in future for that stores management or inventory dept. in real world.

2) The source code of your program

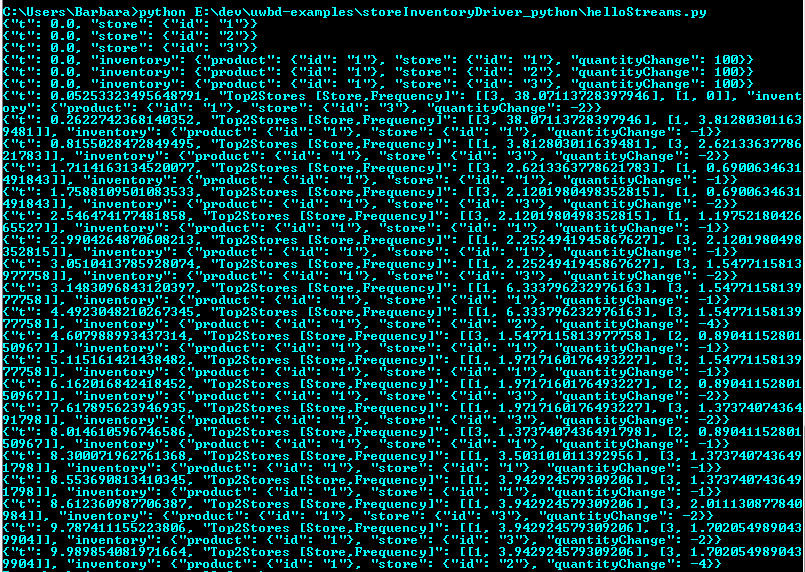
Included helloStreams.py in zip file.

3) How to run your program

All code must be placed in the directory structure as provided by the teacher. I ran the helloStreams.py directly in pycharm using the run option. This code can also be executed at the command line by running the python interpreter and providing full path and file name to the helloStreams.py program. Python.exe 3.x or higher needs to be in your path for windows. Running in linux should be similar but I did not run this program on my linux server.

From Windows CMD window example

python E:\dev\uwbd-examples\storeInventoryDriver\_python\helloStreams.py



4) Output of your program

Included in zip with the name Output-helloStreams-PY.txt

Sample data run with –n 200

From program helloStreams.py which calls a jar simulator

proc = Popen(['java', '-jar', afileSimulator, '-n', '200'], stdout=PIPE, stderr=PIPE)