Project Report for OrderStream automation

Jason Eckstein

[j\_eckstein@outlook.com](mailto:j_eckstein@outlook.com)

1. Project summary

The OrderStream automation script can be used to process open orders and subsequently invoice orders. These are specific functions of the OrderStream website and therefore have been tailored to the interface. The selenium library provides functionality to locate html/CSS elements dynamically, so even after layout changes, the script will most likely still function properly. Name changes would break the script.

The script currently has two modes: process open orders and invoicing, and extract merchant IDs. Process and invoice orders is self-explanatory. Merchant IDs relate to the server running OrderStream. In the future, you may begin supplying merchandise from alternate merchants, and to allow the script to dynamically begin processing those orders, all that is needed is to add the new merchant’s ID and shipping method.

Browsers currently supported by this script are Google Chrome, Mozilla Firefox, and Microsoft Edge. The only requirement for a browser to be used is that the Selenium browser driver be located in the same directory as the script.

The script uses a python library called argparse, which allows greater flexibility for command line switches. Additionally, it provides a useful help utility menu, which explains the included functionality. For basic usage simply type “python orderStream.py -h”.

1. Usage

To begin using the script, place the tracking csv file in the same directory as the script, and rename it to ‘tracking.csv’. Next you will need to copy the three selenium browser drivers into the directory as well. If you’re only going to use one browser the others are unnecessary, but provided for completeness. Finally, go to the command line and execute the script. See command examples below. Note some switches use a single dash ‘-‘while others use double ‘--‘.

Note: first command should be to install dependencies

pip install -r requirements.txt OR pip3 install -r requirements.txt

python orderStream.py -h <-- try this next to see the help menu

python orderStream.py -u username -p password

python orderStream.py -u username -p password -browser Firefox

python orderStream.py -u username -p password --noreview

python orderStream.py -u username -p password --extract

1. Future maintenance

There are a few items that can easily be modified within the script to adjust for future maintenance needs. First is the ‘merchantPids’ list variable located in the global definitions section. As noted by its comment, after you perform the extract function of the script, you can simply paste the formatted output in between this variable’s brackets to enable addition of future merchants.

example:

merchantPids     = ['bedbath', 'snbc'] # when updating id's replace everything inside of square brackets

the string inside the square brackets ‘bedbath’, ‘snbc’ could be replaced with the following ‘bedbath’, ‘snbc’, ‘belk’

In addition to the merchant id, a shipping method code is need to fill the shipping drop down box. This can easily be identified by inspecting the html. Navigate to the open orders page and right click anywhere in the page, and select ‘inspect’. This will open the developer console. Next, right-click the shipping drop-down box, and the developer console will automatically navigate to that element. Expand the element using the tiny plus box, and you’ll see many drop-down options. Find the correct option, and copy its value. Some examples are shown below:

merchShipMeths   = {'bedbath': 'FEDX', 'snbc': 'UG'}#, 'belk': 'FXSP'}

Currently, bed bath and shophq are in use. Their shipping methods are fedex ground (FEDX) and UPS ground (UG). In the comment, Belk’s method is Fedex smart post (FXSP).