Extracting MRN numbers from Hospital folder numbers using Dieter's JavaScript/Puppeteer extract tools

Introduction

Web scraping with these scripts uses the web scraping module called "Puppeteer". This is a module within JavaScript which has a standalone Chromium web browser (can be run from a flash drive) and is controllable with Node (a JavaScript V8 Engine).

The language these scripts are written in, is JavaScript. This is not Java.

Python has a similar Web Scraping module called Selenium. There are other python packages which are smaller and faster: BeautifulSoup, LXML, Python Requests, Scrapy, Urllib and MechanicalSoup to name a few, but each has its advantages and disadvantages.

The main reason I'm using Puppeteer is because I have already learnt some JavaScript, and it opens a web page in a similar fashion as a human would, hence there's little chance that any network traffic will be blocked. It also has an option to see the web page being opened (headless:false mode). The downside is that the whole page needs to load before the data can be obtained, which is not always necessary with other Python Packages (excluding Selenium).

Step 1 – Dependencies:

- Make sure **Node.js** is installed on the computer where you are working (**LTS version** recommended).
- Make sure **VSCode** (Visual Studio Code) is installed on the computer where you are working.

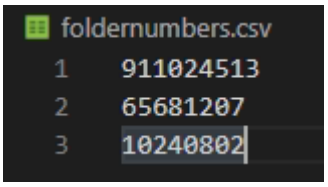
Step 2 – Getting the files and folders ready (can be put on a flash drive):

Copy the Folder called "Web Scraping" anywhere with at least the following files and folders in it:

1. package.json
    a. This file houses the names of the main packages installed in the node_modules folder
2. package-lock.json
    a. This file houses the names of all the branches and dependencies of the main packages
3. node_modules
    a. This folder contains, amongst others, the standalone Chromium browser with the Puppeteer module in it – the biggest module ~400mb)
4. config.json (must be **edited**)
    a. with your own username(s) and password(s)
    b. contains the variables which will often need to be read or edited, like passwords and filenames.

For Scraping MRN numbers from Hospital Folder Numbers (can be done from any network which has access to TrakCare Webview):

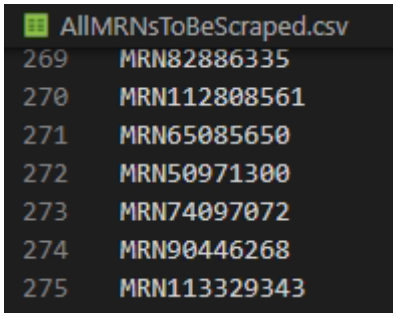
5. getMRNs.js (must be copied)
    a. This is the main script to scrape MRN numbers.
6. foldernumbers.csv -must be formatted as:



This file should contain the folder numbers which you wish to get the MRN numbers for.

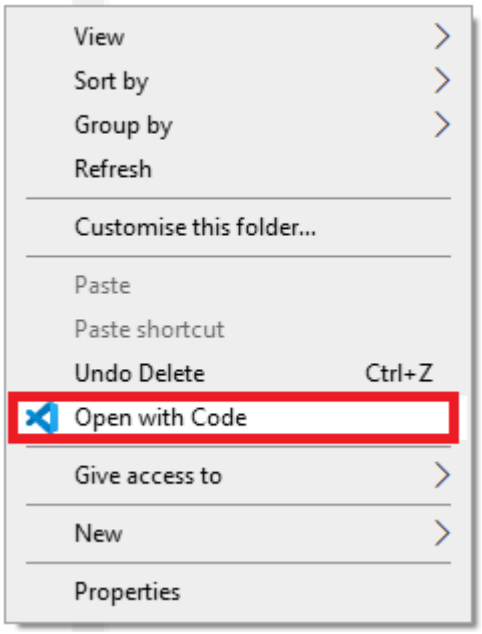
For Scraping the data from the MRN numbers (must be done from within the NHLS – and preferably afterhours, especially if the internet is as slow as it currently is, to prevent this script from pulling data continually through the network)

7. scrapeHST.js
    a. This is the main script to scrape the data from each MRN number.
8. AllMRNsToBeScraped.csv -must formatted like such:



    a. This is the file which should contain the MRN numbers of which you want the data from.

Step 3 – Open the folder in File Explorer by right clicking in an empty space and select "Open with Code"



OR

Open VSCode and open the folder "Web Scraping" within VSCode.

From within VSCode you will see all the files and folders in the left-hand panel and if you click on each, it will open in a new tab in the main window.

Open a new terminal window in the folder: Ctrl + Shift + ` or click "Terminal > New Terminal"

A Terminal window should now be displayed in the bottom panel.

Step 4 : Terminal window commands

In a terminal window type the following command to start extracting:

node getMRNs.js

Chromium browser should launch or an output should become visible in the command line.

The MRN's will output to a raw file: rawwritefile.csv. This file can be opened in VSCode or with any text editor, or saved as .csv by changing the file suffix. If the MRN list has been obtained and cleaned up, save it as "AllMRNsToBeScraped.csv" as noted above.

Then in the terminal window type:

node scrapeHST.js

To let the scripts extract in the background without showing the Chromium browser (headless mode), edit the getMRNs.js or scrapeHST.js files and change "headless:false" to "headless:true".

To stop the extract mid-extract, click on the command window and hit Ctrl+C.