Scraper - Grays Laptops

The purpose of this scraper is to get selected details of laptops that have been sold or are currently being sold on the online auction site [www.grays.com](http://www.grays.com). Following is an example of the web pages, showing the details I would like captured. There are two different types of info to scrape from the site – Laptop Details, and Bidding Details.

# Laptop Details

The following columns will be needed for the **Laptop Details** worksheet (see the attached spreadsheet Laptops.xlsx)

* **Closed** (sourced from **3** in the Laptop Details image above)
* **Auction** (this is the unique Id for a specific auction, and is obtained from the input config file)
* **Lot** (this is the unique Id for a single laptop that is being auctioned, and is obtained from the input config file)
* **Part** (from **5**)
* **Item Name** (from **1**)
* **Bid** (from **2**)
* **Condition** (from **3**)
* **Buyers** **Premium** (from **3**)
* **GST** (from **3**)
* **Warranty** (from **3**)
* **Deliver** **To** (from **3**)
* **Processor** (from **5**)
* **Memory** (from **5**)
* **Storage** (from **5**)

Note that I think the columns: **Part**, **Processor**, **Memory** and **Storage** will need a bit of logic to find, as the Specification dot points on the website are not always in a consistent order. (Refer to the Input section below for how I think you can identify these details).

# Bidding Details

The following columns will be needed for the **Bidding Details** worksheet (see the attached spreadsheet Laptops.xlsx)

* **Auction** (sourced from the input config file)
* **Lot** (sourced from the input config file)
* **Part** (sourced from **5** in the Laptop Details image earlier)
* **Bid** (this is just an integer counter for each row in the bidding history table, where the top row = 1)
* **Bidding** **Details** (sourced from column 1 in the Bidding Details image above)
* **Bid** **Time** (sourced from column 2)
* **Bid** **Price** (sourced from column 3)
* **Bid** **Qty** (sourced from column 4)
* **Win** **Qty** (sourced from column 5)

Note that there will be instances where this table shows no data rows, for example where no bids have yet been placed.

# Input

I think we’ll need an input config file that contains the following info:

**Auction & Lot Details:**

2182381, 1, 100

2182305, 1, 100

2182306, 1, 100

2182307, 1, 100

2182384, 1, 100

2182380, 1, 100

2182385, 1, 100

2182381, 1, 100

2182504, 1, 100

**Processor:** Intel, core, AMD, Ryzen, Xeon, Qualcomm, Snapdragon, CPU, Pentium

**Memory:** RAM, DDR

**Storage:** Hard Drive, Hard Drives, Storage, SSD, Solid State Drive, Solid State Drives

**Part:** Part Number, Part No, Part Num

## Auction & Lot Details:

I have shown multiple rows of parameters in the config file for **Auction & Lot Details** because I would like to gather details for multiple auctions (or just a single auction) in each execution of the script. From the first parameter row:

* **2182381** is an Auction Id
* **1** is the first lot number to scrape
* **100** is the last lot number to scrape

The URL for lot **1** in Auction **2182381**, is: [www.grays.com/lot/0001-2182381](http://www.grays.com/lot/0001-2182381)

## Processor:

Because the **Processor**, **Memory**, **Storage**, and **Part** details that I need are contained within a set of dot points that don’t have a consistent order, I think you may need to search each dot point for various character strings to identify if the dot point represents (for example) the Processor details. The parameter “Intel, core, AMD, Ryzen, Xeon, Qualcomm, Snapdragon, CPU, Pentium” shows a series of ‘words’ that I believe will only be contained within the dot point showing the details for the processor. If there are multiple dot points containing any of these ‘words’, then feel free to concatenate the multiple strings into the one **Processor** column.

As above for **Memory**, **Storage** and **Part**.

However, if you can think of a better way, feel free to use your approach.

# Output

I would like the output written to the spreadsheet (Laptops.xlsx) if this is simple enough to do. Let me know if that is a lot of effort and we can instead use csv format for output.

# Exceptions

I want to scrape the site for auctions that are open, as well as auctions that are closed. Following is an image from an auction that is still underway (Auction Id 2182504). There are a couple of differences, for example:

* Under the **Item** **Name**, an open auction shows “**Closes: x hours, y minutes**”
* Above the **Condition**, an open auction doesn’t show the **Closed** details

