**CS3300 Extra Credit Project:**

**EBAY WEB SCRAPING USING JSOUP**

**Student: Cuong Phan**

**Instructor: Cyril Harris**

**1. About the project:**

The program is an automated web scraping which specifically designed for ebay.com. The purpose of the program is to scrape the product information and display that information into a GUI interface that the user can interact such as read and copy those data.

**2. Main features:**

This experimental version of the program features a simple GUI. This GUI has a textbox for the user to input the product that he/she wants to scrape.

When user hits GO button, the program will then proceed to scrape for the information regarding that product, including the title of the product, its URL, rating, and the number of rates.

After that, program will then display the search results into a JTextPane field.

Users can also use some extra buttons like “NEXT PAGE”, “PREVIOUS PAGE” or “CLEAR”.

“NEXT PAGE” will have the program to scrape the next page of the Ebay’s search results page, whilst “PREVIOUS PAGE” will go back to the previous page. “CLEAR” will just clear all the fields.

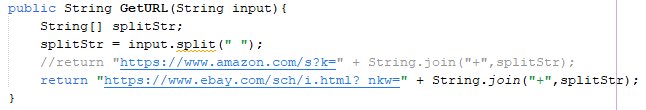
Graphical user interface, application

Description automatically generated

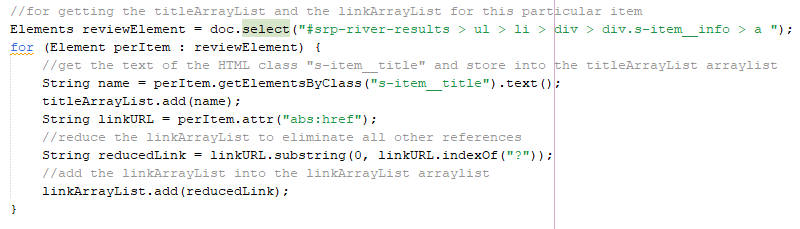
**3. How the program works**

Program has two methods to handle different tasks:

* The GetURL method: this method will split the text string input into multiple unigrams. Then it will connect them with a “+” sign. Finally, program will add the <https://www.ebay.com/sch/i.html?_nkw=> string before that to make it a fully search URL.



* The performScraping method: this is the core mechanism of the program. This method will initialize a doc object of the Jsoup library that will connect to the aforementioned URL. The method will scan through that doc object and search for the title of the item, the link of the item, rating, and number of rates. That information will then be saved into 4 different ArrayLists:
  + titleArrayList: stores the name of the item.
  + linkArrayList: stores the link.



* + ratingCountArrayList: stores the count of rating.

A picture containing text

Description automatically generated

* + rateArrayList: stores the actual rating of the item.

A picture containing text

Description automatically generated

Based on the HTML structure of the ebay website, for the name and the link of the item, program will search inside of "#srp-river-results > ul > li > div > div.s-item\_\_info > a ", while the number of rates and the rating will be searched for inside of "#srp-river-results > ul > li > div > div.s-item\_\_info > div.s-item\_\_reviews > a > div > span" and "#srp-river-results > ul > li > div > div.s-item\_\_info > div.s-item\_\_reviews > a > span > span".

After getting all those information, program will list out all that 4 ArrayLists into the JTextPane field.

Graphical user interface, application

Description automatically generated

If user presses the “NEXT PAGE” button, program will scrape the next result page.

Graphical user interface, text, application

Description automatically generated

The “PREVIOUS PAGE” instead will roll back to the previous search page.

**4. How to use the program – for the Users:**

First, users have to install Jsoup library (jsoup.org). There are multiple ways to do this:

* Add a Dependency into the POM’s <dependencies> section.

Graphical user interface, text, letter

Description automatically generated

* Manually download the “jsoup-1.14.3.jar” from the website and add that into the project.

After that, users can proceed to compile the code and input the product that he/she wants to search for into the txtInput field and hit GO button.

**5. What I learned about the project**

I learned a lot while doing this small project. I knew how to use API to do web scraping, how to read the structure of an HTML page, how to extract data from an HTML document, and how to display that data into a GUI. This project is a very good exercise and highly practical.

**6. Problems with Amazon web scraping**

The actual issue with scraping Amazon.com website is that it blocked all the connect requests. The doc object received nothing. To be able to scrape Amazon.com, the Amazon Product Advertising API or similar paid APIs must be used. However, to participate in the PA API, there are several requirements that I simply cannot satisfy:

Graphical user interface, application

Description automatically generated

Hence for this particular project, I opted to scrape the Ebay.com website, which does not block the connect request from Jsoup.