# *Automated Data Acquisition using Microsoft Excel*

*Luis F. Martínez González*
*Master in Computer Science*
*Jeffrey Duffany, Ph.D.*
*Electrical and Computer Engineering & Computer Science Department*
*Polytechnic University of Puerto Rico*

***Abstract*** *— The process of Web Scraping allows the automated extraction of data from an entire website and turn it into a tabular dataset that can be used for reporting and analysis. There are numerous techniques and tools for web scraping and each one of them are better suited for certain scenarios. One of the most popular techniques, is the direct extraction of the data from the markup language within the website. That is the technique that will discussed in this paper and it will be put into practice by developing a tool using Microsoft Excel. For most individuals that regularly wok with data, Microsoft Excel is one of the most important and useful tools they can have when they need to do a quick analysis or visualize data. Microsoft Excel is powered by an extensible framework and that is Visual Basic for Applications, which allows the implementation of robust functionalities in a tool. In this paper, it will be combined the power of Microsoft Excel and its underlying features, and a popular Web Scraping technique to automate the process of web data acquisition and have this data ready to be processed in Microsoft Excel.*

***Key Terms*** *— Data Automation, Microsoft Excel, Web Programming, Web Scraping.*

## INTRODUCTION

To effectively perform data mining research in the web in a timely manner, an automated data acquisition method is often necessary. There are numerous methods for automating the data acquisition step in a web data research project. In this project, different techniques will be described and one in particular will be discussed in depth and put to practice in the implementation of a tool that will describe the process of Web Scraping. Web Scraping is the process of automatically collecting unstructured data, typically in HTML format, from an entire website and converting it into structured data that can be stored in a file or in a database. The data gathered can then be used to online price comparison, contact scraping, weather data monitoring, website change detection, research, web mashup and web data integration.

The most basic method of web scraping is by a human doing it manually by simply copying and pasting the data from a website into an external application or database. This process is obviously not efficient and would not work in websites that contain large amounts of data. There are different techniques to access this unstructured data behind a website and there are many existing tools that implement these techniques to automate the Web Scraping process.

The technique that will discussed in depth in the paper is the one called HTML DOM Parsing. This technique consists of extracting the data from the markup language that is included within the web page. A detailed description is included further in this paper.

To be able to implement the Web Scraping techniques, programming is usually required using a framework that provides the functionality to implement the different operations required by the technique. Microsoft Excel is a popular tool that is mostly used for data analysis, but it is powered by a dynamic framework which is Visual Basic for Applications. This framework has the functionality required to implement the HTML DOM Parsing technique that will used in the tool and it being a component of Microsoft Excel, also allows to take advantage the built in features of Microsoft Excel, which are very powerful.

## WEB SCRAPING

Web Scraping is a technique for used for extracting large amounts of data from websites with the purpose of storing it in a file or a database in a

structured or tabular format. When you visit a website, most of the data visible to the user can only be seen using a web browser. Most websites don't give the option to the user to save a copy of a dataset being displayed in the browser, it can be a list of items in a store, yellow pages' directories, places of interest websites, real estate websites and others. If the user wants to save a copy of this data and the data is contained in multiple pages, copying and pasting the data into a file can be tedious process [1]. That's when the technique of Web Scraping allows the automation of this process by navigating to the website and programmatically extracting the data from within the websites code.

Most Web Scraping techniques involve programmatically navigating through the website just like the users do when using a web browser. The difference is that when the Web Scraping tool has the contents of the website available, it will automatically extract the data from the code and store it in a file or a database for further use.

Basically any content that can be viewed on a web page can be scraped without the need of an Application Programming Interface (API). An API is a set of tools that can be used by external application to access the functionality of an application. With the use of an API, programmatically you can request a website specific data and it will be received in an already structured format. But not all websites provide API functionality and that s when scraping the data from a web site is a good alternative. A factor that can make the Web Scraping process difficult is poorly formed markup or HTML code. This makes the process more difficult but not impossible, with the right techniques and implementation the data it can also be scrapped.

## WEB SCRAPING TECHNIQUES

There are multiple techniques when it comes to implementing the Web Scraping process. Depending on the structure of a webpage some techniques may work better than others to scrape the data. In this paper the technique that will be discussed in depth is the one called DOM Parsing, but there are other techniques that can be used and some require programming or there can be others that are meant to be executed manually by a human such as Copy-Pasting. Table 1 shows some of the techniques used for Web Scraping:

**Table 1**
**Web Scraping Techniques**

| Technique | Description |
|---|---|
| Copy-Pasting | This is the process of a human manually copying the data that wants to save and copying it into a file. |
| HTML DOM Parsing | Document Object Model (DOM) Parsing is the process of examining the tree structure of an HTML code and programmatically extracting the information contained within the different components [2]. |
| HTTP Programming | This technique consists of making HTTP requests to a web server to retrieve dynamic or static information. |
| (Hyper Text Markup Language) HTML Parsing | HTML Parsing is the process of extracting information directly from the HTML code by using techniques to search for patterns within text such as Regular Expressions. |
| Web Scraping Software | This is using commercial software or existing tools to extract the data from a website. |
| Vertical aggregation platforms | Vertical aggregation platforms create and monitor bots that without human interaction will scrape websites based on the information given in a knowledge base. |
| Semantic annotation recognizing | Semantic annotation recognizing makes use of metadata or semantic markups to locate specific data snippets within the HTML code. If semantic markups or annotations are found then they can be used for DOM Parsing [1]. |
| Computer Vision web-page analyzers | This technique uses machine learning to read the content from a web page as a human being might. |
| Text Grepping | Using the python or Perl programming languages, it can be used the UNIX grep command to extract information from web pages [2]. |

## HTML DOM PARSING

The HTML Document Object Model (DOM) is the programming interface that provides the structure to an HTML document and defines how the different components included within the document can be accessed from programs so that the document structure and the contents can be manipulated. The DOM represents the document as structured group of nodes and elements that have attributes and methods. Figure 1 shows the basic structure of the DOM. Essentially; the DOM allows programming languages to connect to the data behind a webpage [3].

The most common method of accessing and modifying the contents of the DOM is using a scripting language such as JavaScript. Using JavaScript, a webpage can be dynamically changed and that is possible because using JavaScript the DOM can be accessed [3]. For example, Table 2 includes some of the methods that JavaScript to reference a specific section of the webpage and modify it by using a unique identifier of the element or the element type. Once a programming language has the capability of referencing the elements in DOM, this means that the data behind the webpage can be scrapped.

**Table 2**
**Methods for accessing the HTML DOM**

| Method | Description |
|---|---|
| document.getElementByClassName(name) | Reference element using its identifier (i.e. class = "item_name") |
| document.getElementsByTagName(name) | Reference object using the tag name (i.e. "td", "ul") |
| document.createElement(name) | Creates a new element |
| parentNode.appendChild(node) | Append element to an existing element |
| element.innerHTML | Returns the HTML embedded within an element |
| element.setAttribute | Adds or modify attributes of existing elements |
| element.getAttribute | Returns the value of a specified attribute of an element |

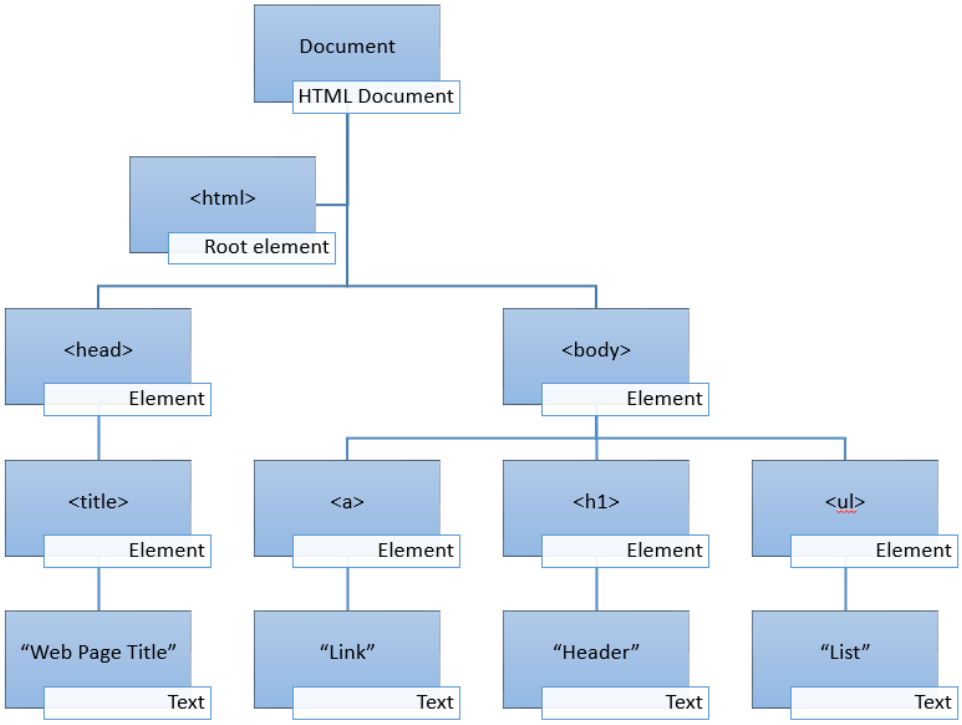


**Figure 1**
**The HTML DOM Tree of Objects**

## IMPLEMENTING A WEB SCRAPING TOOL: EXCEL WEB SCRAPER

As discussed previously, to be able to access the data contained within a website what is needed is a programming language that can navigate through webpages and has the functionality needed to access the HTML document and the Document Object Model (DOM). The DOM combined with JavaScript provide an Application Programming Interface (API) that can be used by external programs to interact with webpages in such ways the it makes the process of Web Scraping possible.

Microsoft Excel is a powerful tool that has many features already built in that allow users to do data manipulation and analysis. It also has a framework that can be used to create custom programs that extend the functionality of Microsoft Excel. This framework is called Visual Basic for Applications and it supports writing code in Visual Basic to implement user defined functions and routines. It is also extensible by using Dynamic Link Libraries (DLL), which allows the developer add functionality that is not already built in [4].

Using VBA and adding a library that extended the framework to include the capability of interacting webpages using VB code, a Web Scraping tool was created. As an example, the tool named "Excel Web Scraper" is pre-configured to scrape data from the popular Point of Interest websites TripAdvisor® and Yelp®. Other websites can be added as long as the fata of interest is located in the HTML of each website respectively.

On most websites, the content is distributed in multiple pages, which means that to be able to extract a complete dataset the code needs to navigate through all the pages and extract the data from DOM from each one. The algorithm used by the Excel Web Scraper to go through each webpage and extract the data is described in Figure 2. The page number is controlled using the URL of the web page as part of the GET HTTP request [5]. In summary, the code loops through every page and within every page it loops through every item to extract the data. Once the data is captured for each item, it is stored in a worksheet in Excel. Once the data is in excel it is ready to be exported to external tools or to be used in Excel for analysis.

```
Initialize page number to zero
Initialize item number to zero
Initialize records per page to 30 (varies)
Initialize page record to zero
while page number is less than record limit
        Navigate to page number webpage
        while page record is less than records per page
                assign to field in excel the value retrieved from the DOM
                add one to page record
        add one to page number
```

**Figure 2**
**Excel Web Scraper Algorithm Pseudo Code**

The number of items per page is a parameter varies depending on the webpage. For example, TripAdvisor® has 30 items per page, while Yelp® has 10 items per page. When defining the loop in the code, these parameters need to be correctly specified for the code to work and extract the correct data.

## SCRAPING THE DATA

The process of scraping the data is most crucial for the functioning of this tool. To be able to extract the data of a given website the following steps had to be taken:

1. Using the Developer tools in Chrome Browser to locate the data in the DOM (see Figure 3)
2. Once the data is located in the DOM, utilize the proper methods provided by the API to extract the data from the DOM (see figure 4)
3. The data may require to be cleaned before it is stored in the worksheet for integrity and validation purposes. To clean it, it was used the built in string manipulation functions in VBA, but it can also be used Regular Expressions to detect the data patterns required.
4. Once the data is cleaned, it is stored in a worksheet for later use.

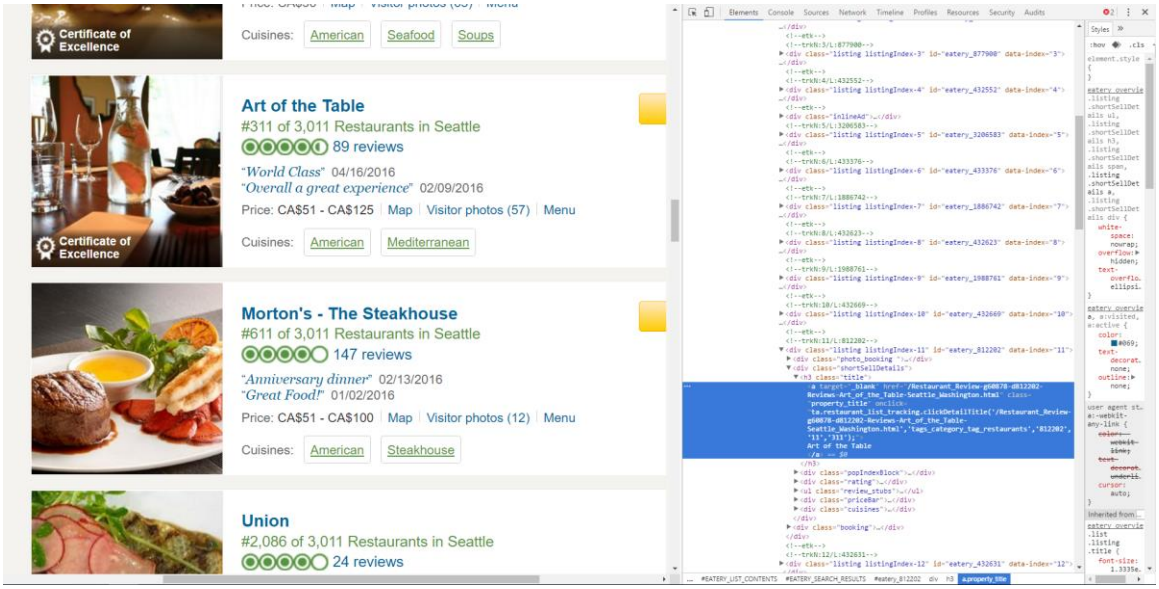


**Figure 3**
**Using Developer Tools to Locate the Data in the DOM**

```
'extract data from HTML
    name = doc.getElementsByClassName("property_title")(cnt).innerText
    rating = Replace(doc.getElementsByClassName("sprite-ratings")(cnt).getAttribute("alt"), " of 5 stars", "")
    reviews = Replace(Replace(doc.getElementsByClassName("reviewCount")(cnt).innerText, " reviews", ""), " review", "")
    price = Replace(doc.getElementsByClassName("price")(cnt).innerText, "Price: ", "")
    sampleReviews = doc.getElementsByClassName("review_stubs")(cnt).innerText
```

**Figure 4**
**Utilizing the API to Extract the Data from the D**

## THE USER INTERFACE

The user interface designed for this tool is simplistic and straight forward. As shown in Figure 5. There are some steps that need to be taken in order to scrape the data from a website. The steps are as follows:

1. Select a Website from the drop down.
2. Select a State and City to look up the data for.
3. Depending on the website selected, different fields will be available to be extracted. Make selections in the list box.
4. Specify the number of records to extract and press the Scrape Data button.
5. Once complete, press the View Data button to see the dataset
6. Optionally, click the Save Data button to export the data to a file.


**Figure 5**
**User Interface**

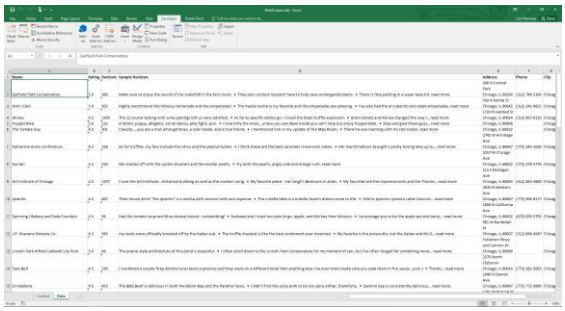

**Figure 6**
**Data Results**

## CONCLUSION AND FUTURE WORK

There is no doubt that Web Scraping is a good alternative to automate the process of data acquisition from the web, especially when websites don't provide the capability of exporting data or an API to programmatically access the data. The technique used to develop the Excel Web Scraper tool, DOM Parsing, it is very powerful and versatile and if implemented in conjunction with a pattern recognition method such as Regular Expressions.

Future works involve enhancing the Excel Web Scraper to use Regular Expressions to automatically try to detect fields of interest within the HTML. That would eliminate or reduce the time required to locate the data in the HTML documents. Also, it would be beneficial to try include other Web Scraping techniques such as HTTP programming, which may be possible to automate if the methodology a website uses to make HTTP requests is found.

## REFERENCES


[1] Wikipedia. (2016). *Web scraping* [Online]. Available: https://en.wikipedia.org/wiki/Web_scraping.

[2] IGN. (2016). *General techniques used for web scraping Wiki Guide - IGN* [Online]. Available: http://www.ign.com/wikis/general-techniques-used-for-web-scraping.

[3] Mozilla Developer Network. (2016). *Introduction to the DOM* [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction.

[4] Wikipedia. (2016). *Visual Basic for Applications* [Online]. Available: https://en.wikipedia.org/wiki/Visual_Basic_for_Applications.

[5] W3schools.com. (2016). *HTTP Methods GET vs POST* [Online]. Available: http://www.w3schools.com/tags/ref_httpmethods.asp.