DYNAMICS OF MODERN WEB SCRAPING: CLIENT EXPECTATIONS VS TECHNICAL REALITY

Umar Khalid*,
[0000-0002-3693-9859]
Armeen Shahid
[0009-0002-9790-8914]

Scraping Solution Ltd.,
London, United Kingdom

Abstract:

The emergence of the World Wide Web has increased the amount of information that is easily accessible on the Internet. This information could be copied manually or scraped by using automated tools. Web scraping also known as web mining, data extraction, or web harvesting refers to extracting raw data from different websites and converting it into useful information. This information can be used by researchers, business organizations, social scientists, and health care professionals for data analysis, understanding the specific market, finding potential clients, people, or market sentiments, or health care wide spectrum studies. This paper focuses on an overview of the changing dynamics of web scraping, a never-ending marathon of web scraping developers and anti-scraping service providers later this article discusses the lack of understanding among clients who still view web scraping as merely a simple process automation to copy data, overlooks the growing complexities involved in navigating the evolving landscape of AI-driven websites and anti-scraping measures.

Keywords:

Web Scraping, Data Extraction, Client Expectations, Automation, Anti-Bot Systems, Technical Challenges.

INTRODUCTION

Data is an essential part of any research, whether it be academic, marketing, or scientific. It is essential for businesses and organizations as it assists their decision making and especially, currently, most of the data can be found on the internet and is available publicly all around the globe on websites [1]. As the internet becomes an ever-growing repository of information, the need to efficiently gather, process, and analyze this data has grown exponentially. Businesses and organizations often face several challenges in acquiring high- quality and required information. Early web scraping techniques involved basic techniques, such as downloading web pages and using regular expressions or simple scripts to extract data. It was often a manual, semi-automated, or occasionally fully automated process, lacking the sophistication and automation of today. Also, it is difficult to access unstructured data and dynamic content generated by JavaScript or embedded within complex HTML structures.

Correspondence:
Umar Khalid

e-mail:
umarkhalid@scrapingsolution.com

DOI: 10.15308/Sinteza-2025-158-163

Moreover, there were few, if any, legal restrictions or formal policies in place. Bot-blocking or anti-scraping systems were virtually non-existent, and at most, basic IP blockers were used— easily bypassed with minimal effort. Today, traditional data acquisition techniques are not efficient and an unpleasant procedure for real-time data collection is not flexible enough for large amounts of data [2].

Modern web scraping involves automatically extracting large bulks of data from websites using software tools and scripts. Data could be in any form such as text, images, videos, or other forms. Throughout the last decade' web scraping has evolved to an entirely new level. With the advancement in technology. It has become more powerful, automated, versatile, flexible, and reliable. The business and research sectors depend heavily on the quality, accuracy, and comprehensiveness of data. Consequently, client expectations are far beyond the true concepts of web scraping. Their expectations are not aligned with reality or are still trapped in the previous decade. As a result, web scrapers frequently encounter challenges in delivering the required output. These issues stem from the inherently complex nature of web scraping along with dynamic websites, and anti-scraping measures like CAPTCHAs, IP-blocking, rate limiting, and evolving legal constraints around data privacy and intellectual property.

The main objectives of this paper are to shed light on the often-overlooked ground realities of web scraping, a technically strong process but its challenges may impact its effectiveness and efficiency. On the other hand, the paper also delves into the expectations that clients typically have when commissioning web scraping projects.

2. OVERVIEW OF WEB SCRAPING

Web scraping also known as web extraction and web harvesting is a technique to extract both structured and unstructured data from websites and transform it into an organized form that can be stored in databases, or comma-separated values [3]. Web scraping is the method involved with separating or extracting information off the web programmatically and changing it into an organized dataset. It is particularly important in fields such as Business Intelligence, Artificial Intelligence, Data Science, Big Data, Cyber Security, Digital Marketing, Sentiment Analysis, and E-commerce development/management in the modern age. Types of scraping discussed by authors in [4] are: Extracting data with an HTML parser or regular expression matching, the second one is by using an application programming interface referred to as API. Web scraping techniques are broadly used in web indexing, web mining, web data integration and data mining [5].

Website investigation, website crawling, and information getting sorted out are the three essential cycles of internet scraping [6]. Due to the wide number of open devices and libraries that offer productive executions of a significant part of the necessary usefulness, web scraping is a straightforward and pretty simple process in general.

In today's era of big data, the demand for relevant information has driven individuals and companies to extensively use web scraping as a crucial tool for gathering data. Companies that aggregate product availability and offer price comparisons have built billion-dollar businesses on the foundation of web scraping. Similarly, they employ web scraping to provide consumers with valuable insights, highlighting the significant role of web scraping in modern business for enabling competitive analysis, market research, and customer decision-making [7]. While others might do web scraping for their interests such as job hunting or getting hoursing offers [8].

The current scenario of web scraping in the generative AI era is quite blurry but at the same time fast-paced, where companies like Cloudflare, Datadome and others are heavily investing in the development of anti-scraping systems with zero tolerance of allowing any kind of scraper or bot to take any data from webpage, however on the other side, big startups like bring data, Zyte api and Oxylabs have been started in recent years to develop web unblockers and proxy Api's to bypass the system developed by Cloudflare or Datadome, Both continuously working to overcome each other's and selling their solution to web developers and scraping developers, a rate race is still ongoing and their market share increasing so the cost of web scraping and data mining. Therefore, with every effort made, data theft news from highly protected systems hit the news every other day [9] and the legal battles are also part of the news now and then [10].

3. DESIGN OF WEB SCRAPING BOTS

A typical design of a bot comprises three phases: website analysis, web scraping, and data organization. The first is the most critical and challenging while the last one is quite straightforward and repetitive. Even with AI advancements, the initial phase still demands significant human involvement, whereas the subsequent stages are comparatively easier, more automated, and can be AI-assisted.

3.1. WEB ANALYSIS

Basic analysis examines the channel through which data is being transferred between the website and the server. Mostly its either get or post request respond by an HTML, XML, or JSON response. The interaction can be either directly with the website server/database or through an API.

Lately, Modern web analysis also includes a thorough study of the anti-bot services deployed on the website along with its version and the study of recent version updates. This analysis further helps the developer to either develop a function to bypass this blocking service itself or use a relevant third-party service.

3.2. WEB CRAWLING

The phase of developing and running a script or program that automatically browses the website and retrieves the required data. It is comprised of languages such as Python, and R. Web scraping libraries; request, beautiful soup, scrapy, and selenium.

3.3. DATA ORGANIZATION

The collected data need to be stored in an organized manner along with the need for data cleaning and pre-processing. Many programming languages, such as R and Python, contain Natural Language Processing (NLP) libraries and data manipulation functions that are useful for cleaning and organizing data [11].

Web scraping could be done in several languages each offering its unique tools and libraries for facilitating the process [12]. Approaches using Python packages discussed in [13] are Regular Expressions, Beautiful Soup, Scrapy, and Lxml. Along with popular headless browsers, selenium is efficient to be used.

4. CLIENT EXPECTATIONS AND TECHNICAL REALITIES OF WEB SCRAPING

This section focuses on the challenges faced by web scraping developers or scraping service providers, highlighting the issues regarding client expectations and the reality of web scraping along with the technical challenges in web scraping.

The client expects and demands something that exactly matches the requirement or idea they are after, sometimes along with non-understanding or rigid behavior particularly in terms of data elements, comprehensiveness, and speed without even knowing the technical difficulties involved. Therefore, the scrapers encounter various challenges such as finding a reliable source for the data (if not mentioned by the client) reflecting the exact information asked, website structure changes, dynamic content, CAPTCHA, and anti-bot measures that may include IP-rate limiting, and device fingerprint monitoring. Although Web scrapers are determined to deal with anti-scraping measures that focus on protecting the content and preventing unauthorized access to large datasets these anti-bot services always keep on updating and sometimes their way to detect a bot may change every day which starts a rat race of analyzing and finding the solution to bypass these services. These challenges may require a lot of human, financial, and computational resources, and could lead to delays in delivery and impact client expectations.

From the clients' perspective, web scraping is considered a low-cost procedure. However, the cost may escalate with the need for advanced scraping techniques, maintenance, and overcoming the ever-evolving anti-scraping services. Scaling a scraping operation also involves significant technical challenges, such as managing server loads, handling a vast number of requests, and dealing with sophisticated anti-scraping measures. While the client may expect the solution to scale effortlessly as their data needs to grow, the complexities often require additional resources and expertise.

On legal grounds, clients may assume that all web scraping activities are lawful and require no specific permissions. However, considering the legalities, scraping large volumes of web data involves serious challenges, including compliance with data privacy laws, intellectual property regulations, and the terms of service set by websites. Both service providers and clients must ensure that scraping operations are conducted within the bounds of applicable legal frameworks to avoid potential liabilities. While there has been an increase in tools and advancement in technologies that can be used for Web Scraping legality and ethics of data collection from the Web are still “grey areas” [11]. Many websites now also require users to agree to “terms and conditions” that explicitly prohibit data scraping, which increases the challenges of web scrapers [12].

In the following section, I will present two case studies from projects undertaken by Scraping Solution. These examples highlight real-world scenarios where web scraping efforts either struggled, escalated in complexity, or were ultimately cancelled. The case studies reflect the dynamic nature of web scraping, where techni-

cal complexities such as anti-bot mechanisms, evolving site structures and client misconceptions significantly impacted project outcomes. The reason for including these specific cases, where the projects faced significant challenges, is because they offer more impactful insights and help steer the article toward a conclusive understanding of the realities surrounding web scraping.

5. CASE STUDY 1: WEB SCRAPING FOR YAHOO FINANCE

5.1. CLIENT REQUIREMENTS

The client approached Scraping Solution to develop a custom scraper for Yahoo Finance, specifically to scrape articles from the news section on a regular basis. Initial discussions revealed the client's desire for a robust, high-speed scraping solution with minimal technical requirements. They expressed an understanding of the web scraping process, which was considered during client profiling. Client profiling at Scraping Solution is a standardized procedure that involves assessing the client's background, previous project ratings, response rate, and other metrics from freelance platforms.

5.2. TECHNICAL INVESTIGATION

Upon initiating the project, Scraping Solution's established methodology involves checking four key aspects:

1. Connection with the Server:

Yahoo Finance did not have an available API for its news section. Thus, it became evident that a browser automation tool like Selenium or a request-based solution would be needed.

2. Static vs. Dynamic Content:

Yahoo Finance's response was identified as dynamic, meaning the HTML structure varied with each request, potentially altering with geographic location. This led to the realization that the scraper would need to handle dynamic HTML responses and geographic fingerprinting, requiring country-specific IP rotation or VPN usage to maintain consistency.

3. Anti-Bot Protection:

The website employed anti-bot mechanisms, though they were not overly restrictive. A simple IP rotation or VPN setup would suffice for maintaining access to Yahoo Finance's articles without triggering anti-bot blocks.

4. Data Storage Format:

As the project involved scraping article data, the data formatting requirement was relatively simple compared to more complex product scraping tasks.

5.3. CLIENT FEEDBACK AND PROJECT OUTCOME

The research was presented to the client along with a recommended solution that employed Selenium for browser automation, supported by IP rotation or VPNs to mitigate the impact of geographical variation in responses. Despite the technical depth and the low-cost recommendation for VPN or IP rotation services, the client rejected the proposed solution, stating that they were unwilling to use VPNs or browser automation tools.

This led to a significant gap between client expectations and technical realities. The client requested a solution that would work without fingerprint rotation or browser automation—an approach that was technically impossible given the nature of Yahoo Finance's security measures and dynamic content delivery. Despite professional communication and attempts to clarify these complexities, the project ended in dispute and was ultimately cancelled, with extensive unpaid research work already completed.

6. CASE STUDY 2: PRODUCT SCRAPING FOR SHOPIFY

6.1. CLIENT REQUIREMENT

A client engaged Scraping Solution to automate the scraping of product data from their supplier's website and format it into a Shopify-uploadable template. The client also requested a cron job to regularly update new products and price changes twice a week. The goal was complete automation, allowing the client to focus solely on the data without needing manual intervention.

6.2. TECHNICAL INVESTIGATION

The website was initially straightforward, with no significant anti-bot services in place. However, the primary challenge lay in scraping product variants, which varied dynamically across different product categories. Some products featured variants like size, color, or weight, and their prices were embedded in JavaScript, requiring additional interaction (click events) to retrieve the pricing data.

The proposed solution was a hybrid of techniques

1. API and Requests:

The scraper would combine API requests, traditional GET/POST methods, and Selenium automation to handle the dynamic JavaScript-based variant data.

2. Handling Product Variants:

A significant portion of the effort was spent coding around the various product departments, each of which required different techniques for parsing product variants. This involved interacting with dynamic elements on the webpage to extract accurate pricing.

3. Automated Updates:

It was recommended that a Flask API or cron job be implemented to check for updates twice a week. Simple IP rotation was also suggested to avoid potential scraping blocks, ensuring the system remained reliable in the long term.

4. Data Formatting:

The scraped data was to be parsed and formatted into a Shopify-uploadable template. This involved heavy data parsing and ensuring compatibility with Shopify's import format.

5. Ongoing Maintenance:

Scraping Solution also proposed hosting the system on their own servers, ensuring real-time monitoring and quick fixes in case of website changes or system crashes. Regular updates to the script were anticipated, as scraping scripts typically require monthly adjustments based on experience.

6.3. CLIENT FEEDBACK AND PROJECT OUTCOME

The client appreciated the research and detailed solution but ultimately decided not to move forward with the development. The client's company was unable to allocate the necessary budget, even though the ongoing maintenance costs were under $200 per month. This hesitation was attributed to two common misconceptions: the belief that web scraping is a simple copy-paste automation process, and a reluctance to incur ongoing monthly expenses for a task they felt could be handled by an in-house hire.

7. DISCUSSION

The complexities and challenges highlighted in both case studies reflect a larger pattern in the web scraping industry—misconceptions about the technical depth required for effective and scalable scraping solutions. As demonstrated by the clients in the examples, a recurring issue is the reluctance to adopt necessary tools, technologies, and ongoing support systems, stemming from a lack of understanding of the nuances involved in web scraping.

In the first case study, the client's hesitation to utilize essential technologies such as browser automation tools and IP rotation reveals a fundamental gap in comprehension. Web scraping is not simply a copy-paste automation task; it is an intricate process that requires dealing with dynamic content, anti-bot systems and geographical restrictions. The assumption that data can be scraped without addressing these complexities reflects a prevalent issue where clients expect scraping solutions to operate with minimal intervention. Browser automation and IP rotation are often indispensable in bypassing modern anti-bot protections and content variations and any attempt to avoid these tools severely limits the scraper's capability. However, such tools are sometimes viewed as unnecessary add-ons, even though they are critical for achieving the desired outcome.

Similarly in the second case study, the client viewed web scraping as a low-cost process assuming that once the system was developed, it would require little to no ongoing investment. This is another common misunderstanding where clients do not fully grasp that scraping environments are not static. Websites frequently undergo changes in their structure, security measures or data presentation which can cause a scraping script to break. The assumption that scraping requires no continuous input from the developer overlooks the necessity of regular updates, error handling and maintenance. In this case, even though the proposed monthly cost was quite low, the client could not just apprehend the expense as he was viewing the scraping process as a "set it and forget it" operation.

This tendency among SMEs and sole traders to resist ongoing costs for web scraping maintenance often stems from a traditional mindset where automation is expected to function indefinitely without requiring human intervention. In reality, automation in the context of web scraping requires constant adaptation, especially when it comes to dynamic websites, complex data structures and anti-scraping measures such as CAPTCHAs or JavaScript-based protection systems.

The reluctance to invest in regular updates and maintenance may also be tied to a perception that the ongoing costs are unjustified or even exploitative. Small business owners in particular, can view monthly charges as "paying for nothing" or mistakenly believing that once the code is written, the process will run indefinitely without further input from the developer. This notion deduces discounts the technical expertise required to ensure that the scraper continues to function smoothly, avoiding crashes, detecting changes in the target website and ensuring data integrity over time.

Both cases exemplify how the lack of understanding of the technical particulars of web scraping can create conflicts between developers and clients. Whether it is resistance to adopt necessary tools like IP rotation or browser automation or the refusal to accept the need for ongoing maintenance. These issues stem from a failure to recognize that web scraping is far more complex than simple automation. As long as these misconceptions persist, developers will continue to face challenges in bridging the gap between client expectations and the technical realities of web scraping.

Ultimately, web scraping is not just about extracting data, but it is about building a robust, adaptable system that can respond to changes in the environment while continuing to deliver value. And that requires more than a one-time investment; it demands ongoing collaboration and support.

8. CONCLUSION AND FUTURE DYNAMICS

Web scraping has become an essential tool for businesses seeking data-driven insights, maintaining their eCommerce stores or managing their marketing strategies, whether for one- off projects or long-term automation solutions. While many clients may approach scraping as a simple, one-time task but the reality often proves more complex. Even in short-term projects, factors such as dynamic content, anti-scraping measures and evolving web technologies mean that the process requires a more thoughtful and flexible approach. For clients, understanding the nuances of web scraping including the potential need for ongoing support or updates can ensure smoother project execution and better long-term outcomes.

Looking toward the future, it's clear that web scraping will become more challenging. As both bot blockers and scrapers continue to invest massively in human and capital resources, the battle over data access will only grow more complex. With websites adopting increasingly sophisticated anti-scraping techniques, web scraping businesses will need more advanced solutions to extract the data they rely on for AI, eCommerce, marketing and analytics. Although scraping is becoming more technically complex and expensive, it will still remain indispensable in a world that thrives on data-driven decision-making.

REFERENCES

[1] I. S. Almaqbali, "Web scrapping: Data extraction from websites," *J. Student Res.*, 2019.

[2] R. Vording, "Harvesting unstructured data in heterogeneous business environments; exploring modern web scraping technologies," M.S. thesis, Univ. Twente, 2021.

[3] K. Mehta, M. Salvi, R. Dand, V. C. Makharia, and P. Natu, "A comparative study of various approaches to adaptive web scraping," in *Lecture Notes in Electrical Engineering*, vol. 601, Springer, 2020, pp. 1245–1256. doi: 10.1007/978-981-15-1420-3_136.

[4] M. Dogucu and M. Çetinkaya-Rundel, "Web scraping in the statistics and data science curriculum: Challenges and opportunities," *J. Stat. Data Sci. Educ.*, vol. 29, no. S1, pp. S112–S122, 2021, doi: 10.1080/10691898.2020.1787116.

[5] M. S. Parvez, K. S. A. Tasneem, S. S. Rajendra, and K. R. Bodke, "Analysis of different web data extraction techniques," in *Proc. 2018 Int. Conf. Smart City Emerging Technol. (ICSCET)*, Mumbai, India, Jan. 2018, pp. 1–5, doi: 10.1109/ICSCET.2018.8537333.

[6] P. H. Milev, "An assessment of information systems for the indexation and analysis of online publications,*"Spisanie "Biznes Upravlenie"*, 2017.

[7] Y. Neil, *Web Scraping the Easy Way*, Georgia Southern Univ., 2016.

[8] G. Pérez Molano, "An overview of web scraping: Technical aspects and exercises," *PRC Repository*, vol. 8, 2023.

[9] Reuters, "Multiple AI companies bypassing web standard to scrape publisher sites, licensing firm says," *Reuters*, 2024.

[10] T. H. Bureau, "Google sued in the U.S. over data scraping for AI," *The Hindu Bureau*, 2023.

[11] V. Krotov, L. Johnson, and L. Silva, "Tutorial: Legality and ethics of web scraping," *Commun. Assoc. Inf. Syst.*, vol. 47, no. 1, pp. 22, 2020, doi: 10.17705/1CAIS.04724.

[12] A. Luscombe, K. Dick, and K. Walby, "Algorithmic thinking in the public interest: Navigating technical, legal, and ethical hurdles to web scraping in the social sciences," *Qual. Quant.*, vol. 56, no. 3, pp. 1023–1044, 2022, doi: 10.1007/s11135-021-01164-0.

[13] I. Valova, T. Mladenova, G. Kanev, and T. C. Halacheva, "Web scraping - state of the art, techniques and approaches," in *Proc. 31st National Conference with International Participation (TELECOM)*, Svishtov, Bulgaria, Oct. 2023, pp. 1–4, doi: 10.1109/TELECOM59629.2023.10409723.