ScrImmo: A Real-time Web Scraper Monitoring the Belgian Real Estate Market

Barzin, Félix; Yernaux, Gonzague; Vanhoof, Wim

Published in:
Proceedings - 2023 22nd IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2023

DOI:
10.1109/wi-iat59888.2023.00054

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

SCRIMMO: A Real-time Web Scraper Monitoring the Belgian Real Estate Market

1st Félix Barzin
University of Namur, Belgium

2nd Gonzague Yernaux
University of Namur, Belgium
gonzague.yernaux@unamur.be

3rd Wim Vanhoof
University of Namur, Belgium

***Abstract*—Web scraping (or Web crawling), a technique for automated data extraction from websites, has emerged as a valuable tool for scientific research and data analysis. This paper presents a comprehensive exploration of Web scraping, its methodologies and challenges. The discussion revolves around a concrete application, namely the automatic extraction of data concerning the Belgian real estate market. We introduce a real-time Web scraper called SCRIMMO and tailored to collect data from websites containing real estate classified ads. The tool is developed in a continuous iterative process and based on an innovative cloud architecture. The paper also briefly addresses the ethical aspects of Web scraping. By integrating insights from previous research and ethical guidelines, this study provides researchers with a comprehensive understanding of Web scraping and its potential benefits, while promoting responsible and ethical practices in data collection and analysis.**

***Index Terms*—Web scraping, Web crawling, Data extraction, Data gathering, Data analysis**

I. INTRODUCTION

In the past decade, the advent of mass data collection tools, such as Web scraping, has revolutionized data acquisition methods [1].

Web scraping, sometimes called Web crawling, is a technique widely used in data mining and research, involving the extraction and retrieval of data from various websites on the World Wide Web. It enables researchers and analysts to gather large amounts of structured or unstructured data for analysis, modeling, and gaining insights across various domains. Web scraping has become increasingly important in scientific studies, as it provides access to vast amounts of information that would otherwise be challenging to obtain [2].

The approach involves the automated extraction of data from Web pages, typically using specialized software or scripts that simulate human browsing behavior to navigate through websites and extract specific data elements of interest. By leveraging the structure and content of Web pages, Web scraping allows researchers to collect data from diverse sources, including online databases, social media platforms, e-commerce websites, news portals, and more [3].

The extracted data can be utilized for a wide range of purposes, such as conducting market research, monitoring online trends, studying social behavior, analyzing sentiment, tracking changes in pricing and availability, and performing quantitative and qualitative analyses [4]. Additionally, Web scraping can facilitate the creation of comprehensive datasets for training machine learning models, enabling automated classification, prediction, and decision-making processes [2].

However, Web scraping also raises important ethical considerations, including data privacy, intellectual property concerns, and compliance with terms of service set by website owners. Researchers must navigate these ethical challenges by ensuring respect for privacy, adhering to legal frameworks, and being transparent about their intentions with the collected data [5].

This paper aims to demonstrate the feasibility of constructing and implementing a data collection tool capable of gathering extensive data sets. To do this, we develop an application called SCRIMMO, that enables real-time monitoring of real estate market activities in Belgium. This endeavor addresses a crucial gap in traditional data collection methods, which are characterized by sluggishness, high costs, and an inability to track market dynamics promptly [6].

The choice of the real estate market stems from the fact that it constitutes a vital sector within the economy, exerting a significant influence on individuals and societal development. Thus, comprehending and analyzing the behavior of the real estate market assumes critical importance, particularly for policymakers who seek to make well-informed decisions, anticipate trends, and address the population's needs. The real estate domain is thus well-suited to illustrate the potential for subsequent analysis of the data collected by the scraper.

In the following, we focus on elucidating the innovative selection of methodologies and tools employed in SCRIMMO, an advanced Web-based data extraction solution utilizing a JavaScript stack to establish a RESTful API deployed on cloud infrastructure. We emphasize the iterative process of designing the scraper, starting from the identification of suitable data sources, followed by data storage and subsequent human processing. The paper highlights our endeavor to create a particularly resilient application. This is crucial, seeing that it fully depends on the Web and its inherent instability. To address this challenge, we design a modular architecture and employ flexible, adaptable, and user-friendly tools.

By scrutinizing existing research, professional practices, and recent advancements, this paper presents the diverse possibilities that arise from analyzing real estate data extracted through this innovative approach. The validity of the resulting dataset paves the way for various analytical treatments, including data visualization tools [7] and machine learning techniques [8].

979-8-3503-0918-8/23/$31.00 ©2023 IEEE
DOI 10.1109/WI-IAT59888.2023.00054

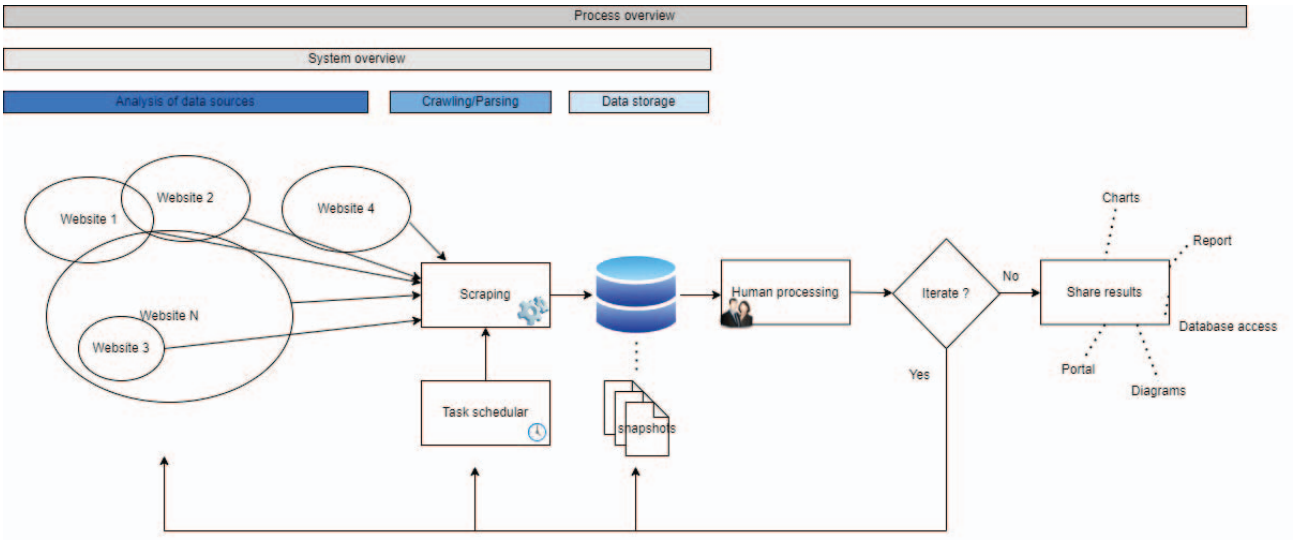

Fig. 1. The iterative process used to design, develop and maintain SCRIMMO

II. ITERATIVE DESIGN PROCESS FOR A SCRAPING SYSTEM

In this section we propose a 7-steps methodology for designing scraping systems. The process is an adaptation of approaches exposed in related work [1], [6], [9]. The resulting framework was used to design SCRIMMO.

The seven steps of our approach are detailed below, and the resulting iterative design process is depicted in Fig. 1.

1) **Selection of Quality Data Sources**
   In this initial step, meticulous consideration is given to choosing one or more reliable data sources.
2) **Analysis of Network Traffic**
   Network traffic analysis allows to identify the constituent elements of the targeted page(s) and to determine the specific properties of interest that will be recorded subsequently. It is typically in this phase that developers identify the HTML tags containing the target data.
3) **Task Scheduler Creation**
   A task scheduler is then developed to facilitate scheduled data harvesting according to the desired frequency. Regular requests are sent to the RESTful API at defined intervals to trigger the data collection process. In SCRIMMO, we used the Microsoft Azure Logic Apps scheduler.
4) **Scraper Development**
   The development phase is rather technical in the sense that the scraper must rely on technologies able to manipulate Web pages and/or Web browsers. For the implementation of our scraper, we used `Express.js`, a lightweight JavaScript framework, to build our API. We also used `Puppeteer.js`, a JavaScript library capable of interacting with a Chromium engine.
5) **Data Storage**
   Extracted data is stored in a non-relational database, specifically MongoDB, to leverage its flexibility. Given the absence of a rigid schema requirement, the ability to store unstructured data in JSON format aligns perfectly with tour project's needs. We chose MongoDB as it allows for simultaneous extraction processes without causing access conflicts.
6) **Human Processing Phase**
   This stage involves eliminating duplicates and extraneous data. Ad-hoc techniques for detecting outliers, data filtering, and ensuring the legitimacy of extracted data are to be devised. Data validation is achieved by comparing the extracted data with information published by official bodies. A comprehensive understanding of the target data and its intended usage is essential. To this end, Exploratory Data Analysis (EDA) techniques can be employed to generate insights, identify trends, and uncover patterns. EDA is typically combined with graphical visualization tools, descriptive statistics, and other segmentation methods that enhance the understanding of the field of interest.
7) **System Refinement and Information Distribution**
   Several iterations of the previous steps are needed for the system to reach a satisfactory state. Decisions can then be made regarding the distribution of the information, namely who it will be made accessible to, and in what form (charts, diagrams, etc.).

The main aspects that differentiate our method from existing approaches such as that of [9] are, first, the fact that we do not conduct a preliminary data analysis before extracting the data; instead, we allow for more than one iterations of the whole process to refine its outputs (including the data analysis) over time– which is of crucial importance in real-time systems. Another distinction is the fact that we do not use a temporary storage before transforming the data; in contrast, we transform the data before storing it. Finally, instead of Notifications and Resource Cleaning steps, we propose a step called Information Distribution.

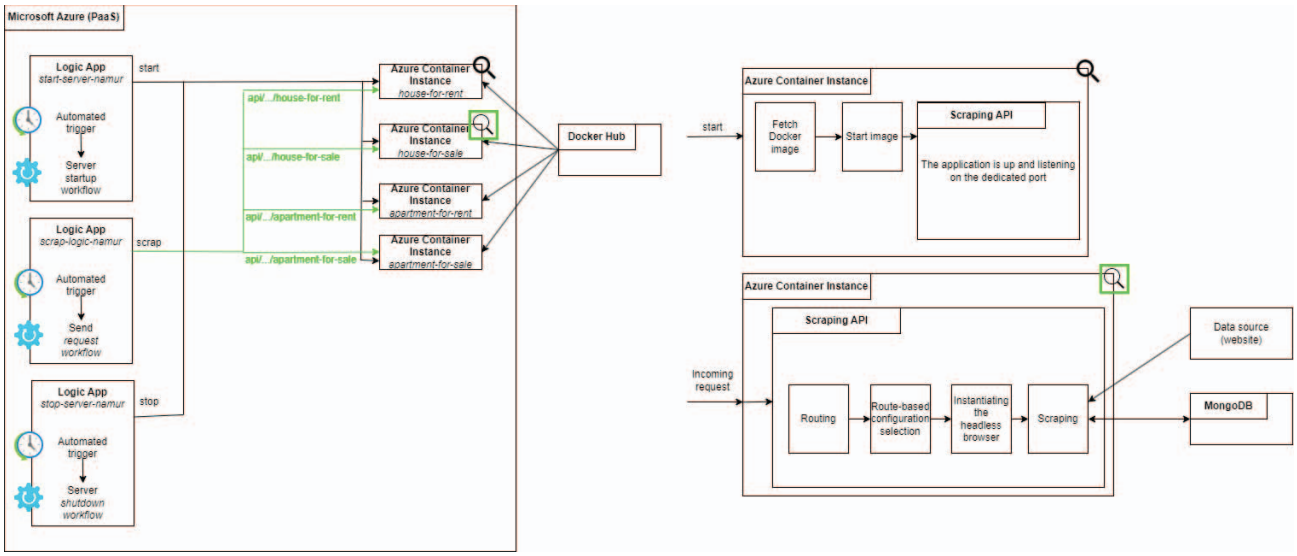

Fig. 2. The cloud architecture used to deploy SCRIMMO

III. CLOUD ARCHITECTURE

Platform as a Service (PaaS) solutions offer global accessibility for running scrapers continuously without the burden of physical infrastructure, maintainability and security concerns.

Fig. 2 illustrates a modular and resilient cloud architecture design, that we used to build our own scraper. Modularity is a key principle in our architecture to facilitate system extension and enhance maintainability. This is particularly important for scrapers, as proactive and regular maintenance is crucial for ensuring smooth operation and longevity of the application, thereby ensuring efficient and reliable data collection [10].

The architecture includes "Logic Apps," which are workflows responsible for distinct tasks such as server activation, triggering harvesting orders, and server deactivation. This breakdown enhances the readability of the no-code configurator through a sequential timeline and enables control over CPU usage time for cost management.

"Container Instances" serve as server equivalents and execute individual scraper instances for data harvesting. We have implemented a one-server-per-data-domain approach, where separate servers are responsible for collecting specific types of data, such as houses for sale or houses for rent. This modularity serves multiple purposes: simplifying debugging by reducing the number of logs to review in case of unexpected failures, controlling CPU usage time by replaying only the failed harvest, enabling parallel execution for performance adjustment, facilitating maintenance, and improving resilience through the tracking of changes along with error monitoring and performance management.

The harvesting process is straightforward. When the server activation workflow is triggered by a recurring task scheduler, it switches on a server (container instance). Upon waking up, the server fetches a Docker image from the previously uploaded scraper on the Docker Hub. Once the image is downloaded and launched, the application becomes operational. As many APIs, it remains idle until it receives an HTTP request.

The second workflow is timed to start a few minutes after the first workflow begins. It sends parameterized HTTP requests to the activated servers. For instance, a request such as `api/.../house-for-rent?province=namur` is directed to a specific server dedicated to collecting data on houses for rent in the Namur province. Upon receiving a parameterized request, the scraper configures itself based on the received parameters and route. It then initializes a headless browser instance, referred to as the "browser object," which provides all the functionality of a Chromium browser. The scraper navigates the data source, extracts relevant information, formats it, and stores it. A final workflow is designed to shut down the servers after a predefined time interval to control CPU usage. This is essential to reduce costs, as the server consumes nearly 1 GB of memory when idle.

SCRIMMO was built according to this architecture and to the principles exposed in the rest of the paper. It is available online[1] for the interested researchers and data analysts.

IV. DATA EXTRACTION AND TRANSFORMATION

The scraping process involves analyzing data sources using a database retroengineering approach. Through an examination of network traffic from the source Web pages, we selectively identified the properties of interest for data extraction. This ensures that the extracted data is formatted, structured, and readily usable. Each stored classified advertisement is enriched with metadata, including information about its source, extraction date, and other relevant details. These additional elements are essential for efficient human processing tasks such as duplicate removal and data filtering.

[1]See `https://github.com/felixbarzin/scrap`. The repository contains an explanatory video as well as installation instructions.

Depending on its assigned domain, the scraper initiates a search query (e.g., classified ads for houses for sale in the province of Namur) and systematically browses through the returned pages to compile a list of URLs for classified ads. Once it reaches the last page, the scraper proceeds to visit each URL stored in the list. At this stage, the scraper verifies whether the current classified ad already exists in our database and determines if it has the clearance to record the ad. If the classified ad is already present, the scraper checks for any modifications. In case of any changes, a copy of the modified version is saved, and the initial version is archived.

The registration stage includes data validity checks to ensure the consistency of the classified ad. For instance, a rental price below 1 euro is likely to indicate a scam or encoding error.

V. Statistics and Data Analysis

As part of the design process, we integrated Exploratory Data Analysis to refine our system and enhance our understanding of the studied domain. We utilized MongoCharts and Tableau Desktop to generate graphical representations based on snapshots from our database.

At the conclusion of our study, we accumulated four months' worth of data. Through chart generation, we can obtain a comprehensive and extensive overview of the real estate market, capturing its activities. By regularly updating the database with daily or weekly snapshots, we enable real-time monitoring of market dynamics.

The analysis possibilities can be expanded to include cross-analysis with other factors such as crime rates and household size growth, among others. Furthermore, incorporating machine learning tools [6, 7] enables diverse combinations and offers informed insights into the current state of the real estate market for planners and other stakeholders.

Note that our study serves as a preliminary exploration and does not claim to uncover definitive patterns and trends in the collected data pertaining to the Belgian real estate market. Therefore, we do not present our own statistical data at this stage. Additionally, our dataset is currently focused on a single province, resulting in a relatively small sample size (a few thousand lines per data domain), which limits the precision of predictive capabilities. To improve data collection, future work could involve harvesting data spanning over a year to employ a machine learning approach. Equipped with such an extended data collection, we will be better positioned to determine whether significant patterns exist within the data to inform real estate decision-making in Belgium.

VI. Ethical considerations

While there is no universal framework for addressing ethical considerations in scraper design, we can apply some fundamental principles of scientific research (as discussed by Rennie et al. [11]). Mancosu and Vegetti [5] notably recommend that scientists be guided by the principles of respect for the individual, beneficence, and justice. In that spirit, ScrImmo adheres to the three following ethical prescriptions:

- Preserve privacy and avoid harming people.
- Respect the legal framework that safeguards individuals.
- Adhere to the terms of service of the platform containing the data.

In particular, as should be the case for any scraper, we do not harm the legitimate interests of the database producer, nor do we employ any deceptive tactics, such as captcha solvers or proxies, to bypass protection measures. We schedule our scripts to run during off-peak hours (between 12am and 5am) and avoid running intensive queries. Being focused on research, we do not seek financial gain from the collected information, nor do we (re)distribute the gathered information.

VII. Conclusions and Future Work

Throughout the paper, we have successfully established an efficient, modular, maintainable, real-time, and scalable scraping system relying on Cloud-friendly technologies, namely Microsoft Azure, `Puppeteer.js`, `Express.js`, GitHub, Docker, DockerHub, MongoDB, MongoCharts and Tableau Desktop. The collected data is comprehensive, reliable, and suitable for analysis. This success opens up possibilities for future work, including expanding the system to cover other regions, integrating alternative data sources such as social networks, and applying machine learning techniques for predictive purposes [12]. We also plan on more formally evaluating the collected data by comparing it with official regional data when available, and by pursuing our EDA. ScrImmo is available online.

References

[1] C. Bertram, G. Betz, P. Ebel, and F. Naumann, "A systematic literature review on methods for web scraping," in *Proceedings of the International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018, pp. 1578–1585.

[2] S. Gupta and P. Dogga, "A survey on web scraping tools and techniques," in *9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2018.

[3] F. Wu, H.-J. Zeng, H. Yan, and W.-Y. Ma, "Understanding and characterizing web content extraction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 6, pp. 760–772, 2006.

[4] T. Venturini, M. Jacomy, and P. Tubaro, "Web scraping as a method for social research," *Sociological Methods & Research*, 2018.

[5] M. Mancosu and F. Vegetti, "What you can scrape and what is right to scrape: A proposal for a tool to collect public facebook data," *Social Media + Society*, vol. 6, pp. 1–11, 2020.

[6] S. Rani and R. Garg, "Web scraping methodology for e-commerce data extraction," in *Proceedings of the 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2017, pp. 1937–1941.

[7] D. Fava, G. Guaragno, and C. Dall'Olio, *Decision Support Systems for Urban Planning*. Springer Open, 2016.

[8] D. Wang and L. Jing, "Mass appraisal models of real estate in the 21st century: A systematic literature review," *Sustainability*, vol. 11, 2019.

[9] P. Milev, "Conceptual approach for development of web scraping application for tracking information," *Economic Alternatives*, pp. 475–485, 2017.

[10] A. Ghatage and M. Shelar, "A comparative study of web scraping tools for extracting structured data from websites," in *5th International Conference on Advanced Computing & Communication Systems*, 2019.

[11] S. Rennie, M. Buchbinder, E. Juengst, L. Brinkley-Rubinstein, C. Blue, and D. Rosen, "Scraping the web for public health gains: Ethical considerations from a 'big data' research project on hiv and incarceration," *Public Health Ethics*, vol. 13, pp. 111–121, 2020.

[12] A. Grybauskas, V. Pilinkienė, and A. Stundžienė, "Predictive analytics using big data for the real estate market during the covid-19 pandemic," *Journal of Big Data*, vol. 8, no. 105, p. 105, 2021.