IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, June 2022
Impact Factor: 6.252

Hybrid Approach for Scraping HTML within JSON Structure

Shivani D. Chaudhari[1] and Sudeshna Roy[2]
Students, Master of Computer Application[1,2]
Bharati Vidyapeeth's Institute of Management and Information Technology, Navi Mumbai, India

Abstract: *Web scraping is a technique for extracting unstructured information from a website and storing it in a structured format. Scraping is an important approach for autonomously retrieving third-party data. For example, If you want to see all of the available transportation on a specific day, you may have to scrape travel websites to gather the information you need. In this research paper, we are attempting to scrape a website that has a JSON structure but also includes an HTML format that must be scraped along with the JSON data. Doing scraping with RegEx for HTML and JSON extraction is a technique for collecting data from these types of websites. To scrape JSON data, we must first get a JSON response and scrape data using key value pairs and then use RegEx to retrieve information from HTML contained within that JSON structure.*

Keywords: JSON, RegEx, Web scraping, Hybrid scraping, scrapy, HTML scraping, Data extraction

I. INTRODUCTION

The technique of obtaining data from web pages is known as web scraping. Some online data is offered in a format that makes it easier to collect and use information, such as downloadable comma-separated values (CSV) datasets that can subsequently be imported into a spreadsheet or loaded into a data analysis tool. Data is frequently not easily available for reuse, regardless of the fact that it is publicly available. It could be in a PDF, a table on a website, or scattered across numerous web pages. Web scraping is used for a variety of purposes, including market research, sales, advertising, finance, and ecommerce. It is frequently used to learn more about your competitors.

Scraping a website to extract information for reuse can be done in a number of ways. In its most basic form, this can be accomplished by copying and pasting snippets from a web page; but, if there is a big amount of data to be extracted, or if it is dispersed across a large number of pages, this may be impractical. Instead, specific tools and techniques can be used to automate this process by specifying which sites to visit, what information to look for, and whether data extraction should stop when a page's end is reached or continue following hyperlinks and continue the process recursively. Web scraping can also be automated to determine if the procedure should be repeated at regular intervals and to catch data changes. Web scraping can be done in two ways: one by visiting www via HTTP via a web browser, and the other by using a bot or web crawler.

Scraping the website, on the other hand, is unique, and every website has its own manner of populating data on the page. HTML, API calls, and XML formats are only a few possibilities. But what if the data on the website is populated using a variety of formats? In the research paper, there is one case that is examined. It is a unique way in which a HTML is embedded within the API call. The data will be extracted via the API, and the HTML data will be retrieved. To scrape this type of website, we'll use Python to extract data using key value pairs, followed by RegEx to parse HTML data that is present as a value in one of the keys.

1.1 Abbreviations and Acronyms

	Abbreviations and Acronyms
API	Application Programming Interface
RegEx	Regular Expression
HTML	Hypertext Markup Language
XML	Extensible Markup Language
DOM	Document Object Model
JSON	JavaScript Object Notation

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, June 2022
Impact Factor: 6.252

II. LITERATURE REVIEW

Before we get started on the paper, let's go through basic web scraping strategies.

A. HTML Parsing

- The HTML parser is a tool for processing structured markup. It creates the HTMLParser class, which is responsible for parsing HTML files.
- JavaScript is used to parse HTML pages that are either linear or nested. It is a method for resource extraction, screen scraping, and text extraction that is quick and reliable.

B. DOM Parsing

- The Document Object Model is a style structure and content specification for XML files.
- Scrapers employ DOM parsers to gain a detailed look at the structure of a web page. They can also employ a DOM parser to extract nodes holding information, and subsequently scrape web pages using a tool like XPath.
- The full web page or simply parts of it can be extracted using the Firefox or Internet Explorer browsers.

C. XPATH

- A query language for XML documents is XML Path Language. Because of its tree-like structure, XPath may be used to browse XML documents by choosing nodes using various parameters.
- To scrape a complete web page, XPath can be combined with DOM parsing.
- XPath is a technology that selects nodes or node-sets in an XML text using path expressions.
- Even though it is not a programming language, it allows you to define an expression that points to a specific HTML element or tag attribute without having to manually crawl over any element lists.

D. Text pattern matching (RegEx)

- RegEx is a pattern matching mechanism that matches certain patterns based on provided combinations which can be used as filters to obtain the desired output.
- RegEx may validate any character combination, including special characters such as line breaks.
- Regular Expressions are a type of expression that can be used in any programming language.

III. METHODOLOGY

Obtaining web resources and then extracting necessary information from the acquired data are the two consecutive processes in the process of scraping data from the Internet. A web scraping method begins by composing an HTTP request to obtain resources from a certain website. This request can take the form of a URL with a GET request or a POST request in an HTTP response. The requested resource will be retrieved from the website and then sent back to the given web scraping engine once the request has been successfully received and processed by the targeted website.

With the usage of an API, a website can sometimes make it easier for a user to have direct access to their data. This essentially means that the business has created a set of specialized URLs that deliver this information in its purest form that is without any presentation formatting. This unprocessed data is frequently in JSON format, which we may read and retrieve using Python. When it comes to websites that don't provide API data, all we have to do is scrape the information using HTML. The majority of web scraping is done with HTML.There are several methods for scraping data which includes DOM parsing, XPath, RegEx, and CSS selectors. In almost all circumstances, a small sample from a huge file is all that is necessary for example pricing information from an ecommerce page. As a result, looking through an HTML document for the correct information is an important aspect of scraping.

In this research we will look at one scenario in which the response contains JSON data as well as HTML data embedded within that JSON format

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, June 2022
Impact Factor: 6.252

```
data : {
      arrTime: "1400"
      bagFilterCheckboxes: "no_baggage"
      comfortFilterCheckboxes: ["sms_ticket"]
      depTime: "0600"
      html: "\t\n\n\n<div
      id=\"185d740221d8ca92ba61208dfef74e51\"\ndata-route-currency=\"CZK\"
      data-route-name=\"<i class='mapshow icon'></i>\ndata-reserve=\"\"
      \ndata-arrival-time-stations=\"\"
      \ndata-arrival-stations=\"\"\ndata-departure-time-stations=\"\"
      \ndata-departure-stations=\"\" \ndata-max_seats=\"10\"
      \ndata-info-step=\"0\"\ndata-free_seats=\"1\"\ndata-allow-transfer=\"0\"
      data-price_fltr=\"695.52\" data-distance=\"28800\" data-rating=\"4\"
      data-bonus=\"42\" data-transporter=\"Cc21b3b6dd13591c404399a36a5dfcb4bC\"
      data-deptime_fltr=\"0600\" data-arrtime_fltr=\"1400\"</div>"
      linkFilterCheckboxes: "hasNoLink"
      price: "695.52"
      rating: 4
      transporterFilterCheckboxes: "Cc21b3b6dd13591c404399a36a5dfcb4bC"
      }
```

Fig. 1. Reference Code to demonstrate html content present inside Json structure.

"Fig. 1" is an example of a one-of-a-kind scenario in which we have JSON data that we obtained from an API, and one of the parameters contains the html of that website. As you can see in the diagram, JSON data is made up of key-value pairs, much like a Python dictionary. In fact, we'll put it into a python dictionary object so that we can scan through it and retrieve the information we need.

In this case, RegEx is used to scrape the HTML content included within the JSON format. Regular expressions work by being given a pattern to work with. After that, the content is matched to the pattern, and an output is generated. [4]RegEx has been used in software for a variety of purposes, including JavaScript validation in forms, text encoding in a specific pattern for development purposes, and decrypting encoded portions in database entries. Despite these evident advantages, using RegEx requires users to have a thorough understanding of its physics in order to execute it efficiently.

The process flow for implementing the solution is outlined below in "Fig. 2".

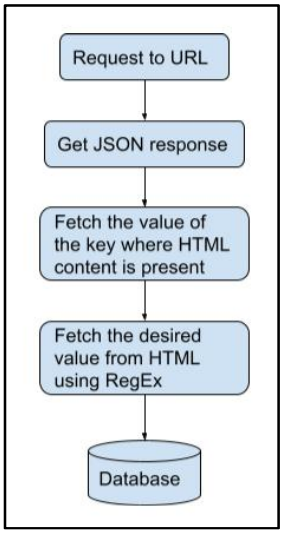

Fig. 2. Flow chart

Following are the steps to scrape the website having "Fig. 1" as a pattern to populate its data:

Step 1: We will be using the scrapy framework and for that we have to install scrapy and further start a new project named DemoScraper.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, June 2022
Impact Factor: 6.252

> pip install scrapy
> scrapy startproject DemoScraper

Step 2: Below is a simple spider named ”demo” that was generated within the DemoScraper project. We pass the URL of the website to be scraped into the scraper, and the request is then yielded by adding the appropriate parameters which are required to scrape the website and further the request is passed to the parse function. We will be importing the “re” library so that we can scrape the HTML present inside the JSON structure.

```
import scrapy
import re
class DemoScraper(scrapy.Spider): name = "demo"
def start_requests(self): urls = [
’https://demowebsite.com/script/’,
]
for url in urls:
yield scrapy.Request(url=url, callback=self.parse)
def parse(self, response): page = response.body
```

Step 3: Inside the parse function, the main scraping starts. So as to scrape the JSON structure, load that data first and for that we will be doing the following.

```
data = json.loads(page)
```

Step 4: After loading the data we will fetch the required values which are present in JSON as mentioned in “Fig. 1”. For example below we are fetching value for price and rating.

```
price = data[‘price’] rating = data[‘rating’]
```

Step 5: To scrape the HTML part, we will be storing HTML data into variable.

```
html_data = data[‘html’]
```

Step 6: RegEx will then be used to scrape HTML data. For example, we’ll scrape a value called “data-distance” from HTML. Syntax to scrape using RegEx is as follows.

1) Function used: re.search()
2) Parameters: Matching RegEx string & Data to match.
3) Group the function according to your requirement.

```
distance = re.search(‘data-distance= "(.*?)"’,html_data).group(1)
```

Step 6: Finally we will store the data into the database.

IV. CONCLUSION

In this study, we used one unique website as an illustration of how the hybrid strategy can be employed. Other types of websites may necessitate the use of two or more scraping strategies.

REFERENCES

[1] Zhao, Bo. ”Web scraping.” Encyclopedia of big data (2017): 1-3.
[2] Gunawan, R., Rahmatulloh, A., Darmawan, I., & Firdaus, F. (2019, March). Comparison of web scraping techniques: regular expression, HTML DOM and Xpath. In International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018) Comparison (Vol. 2, pp. 283-287).
[3] Chapagain, A. (2019). Hands-On Web Scraping with Python: Perform advanced scraping operations using various Python libraries and tools such as Selenium, RegEx, and others. Packt Publishing Ltd.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, June 2022
Impact Factor: 6.252

[4] Morsidi, F., Sulaiman, S., Wahid, R. A., & Malim, T. (2017). Feature extraction using regular expression in detecting proper noun for Malay news articles based on KNN algorithm. Journal of Fundamental and Applied Sciences, 9(5S), 210-231.

[5] Uzun, E. (2020). A novel web scraping approach using the additional information obtained from web pages. IEEE Access, 8, 61726-61740.

[6] Ferrara, E., De Meo, P., Fiumara, G., & Baumgartner, R. (2014). Web data extraction, applications and techniques: A survey. Knowledge-based systems, 70, 301-323.

[7] Uzun, E., Agun, H. V., & Yerlikaya, T. (2013). A hybrid approach for extracting informative content from web pages. Information Processing & Management, 49(4), 928-944.

[8] Uzun, E., Serdar Gu¨ner, E., Kılıc¸aslan, Y., Yerlikaya, T., & Agun, H. V. (2014). An effective and efficient Web content extractor for optimizing the crawling process. Software: Practice and Experience, 44(10), 1181- 1199.

[9] Wu, Y. C. (2016). Language independent web news extraction system based on text detection framework. Infomation Sciences, 342, 132-149.

[10] Kumar, M., Bhatia, R., & Rattan, D. (2017). A survey of Web crawlers for information retrieval. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7(6), e1218.

[11] Song, D., Sun, F., & Liao, L. (2015). A hybrid approach for content extraction with text density and visual importance of DOM nodes.
Knowledge and Information Systems, 42(1), 75-96

[12] Qureshi, P. A. R., & Memon, N. (2012). Hybrid model of content extraction. Journal of Computer and System Sciences, 78(4), 1248-1257.