**How to Use ChatGPT for Web Scraping**

How to scrape websites using ChatGPT, the challenges involved, and how to overcome them.

[[Marco Rodrigues](https://medium.com/@macrodrigues?source=post_page-----29ce04818e72--------------------------------)](https://medium.com/@macrodrigues?source=post_page-----29ce04818e72--------------------------------)

[[Python in Plain English](https://python.plainenglish.io/?source=post_page-----29ce04818e72--------------------------------)](https://python.plainenglish.io/?source=post_page-----29ce04818e72--------------------------------)

[Marco Rodrigues](https://medium.com/@macrodrigues?source=post_page-----29ce04818e72--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2Ff2910ec37c18&operation=register&redirect=https%3A%2F%2Fpython.plainenglish.io%2Fhow-to-use-chatgpt-for-web-scraping-29ce04818e72&user=Marco+Rodrigues&userId=f2910ec37c18&source=post_page-f2910ec37c18----29ce04818e72---------------------post_header-----------)

Published in

[Python in Plain English](https://python.plainenglish.io/?source=post_page-----29ce04818e72--------------------------------)

·

12 min read

·

Dec 8, 2023

565

3

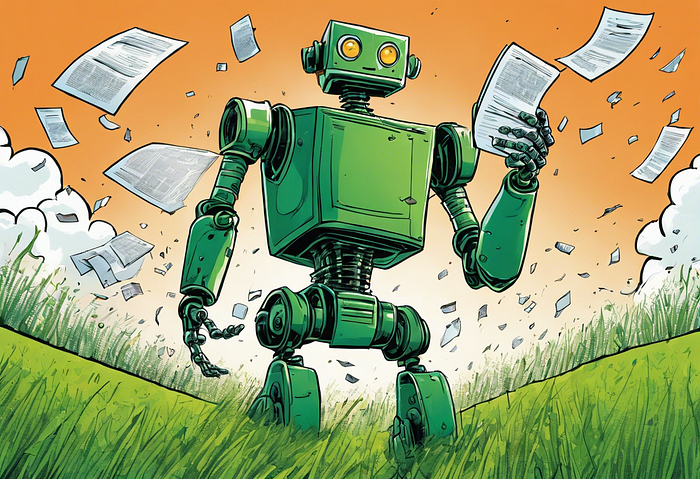


Image generated with [DreamStudio](https://beta.dreamstudio.ai/" \t "_blank)

The dawn of AI has revolutionized the way developers approach their work, especially with LLMs (Large Language Models) and [ChatGPT](https://chat.openai.com/). This versatile tool, developed by [OpenAI](https://openai.com/), offers two versions: a free one powered by the GPT3.5 LLM model and a paid tier with enhanced features, including plugins and access to the more advanced GPT4 model. Both models are trained on massive amounts of data, but regardless of the version, ChatGPT serves as a swiss-army-knife for developers, tackling a wide range of tasks.

ChatGPT’s capabilities extend to various domains, enabling rapid task completion and code debugging. One such domain is web scraping, where it streamlines data extraction processes. With this tool, web scrapers can embed HTML elements in their prompts and specify which library to use as a parser, like [Beautiful Soup](https://www.crummy.com/software/BeautifulSoup/bs4/doc/) and [Scrapy](https://scrapy.org/). For more advanced web scraping, developers can also ask for headless browser automation tasks with [Selenium](https://www.selenium.dev/) and [Playwright](https://playwright.dev/python/) for instance.

In this article, you will learn how you can use ChatGPT to quickly develop fully functional web scrapers. We’ll also look into some challenges and how to make use of an easily integrated [comprehensive solution](https://tracking.nimbleway.com/SH20) to get around these blocks. But first, let’s see how ChatGPT can help speed up the data collection process.

**How to Scrape Websites Using ChatGPT**

Traditionally, web scraping can be a time-consuming task, demanding expertise in HTML parsing libraries and automation tools. ChatGPT makes this process much more user-friendly. Simply provide a basic prompt specifying the website to scrape and the desired technology, and the GPT model will generate a structured script as a starting point.

With further prompts, users can craft robust scripts without extensive web scraping knowledge. The paid version expands upon this functionality with several plugins such as [Scraper](https://gptstore.ai/plugins/api-gafo-tech), tailored specifically for advanced scraping projects. The quality of the outputs ultimately depends on one’s ability to write concise prompts. You can provide complex instructions or multi-step queries in a conversational manner, which is helpful when dealing with websites that require several interactions to retrieve the desired information.

Along with that, ChatGPT also excels in providing recommendations and step-by-step guides to accomplish your web scraping goals.

In this article, we’ll demonstrate how to use ChatGPT for web scraping by scraping [Walmart](https://www.walmart.com/).

However, for all its benefits, it’s worth bearing in mind that there are still challenges you will encounter when trying to scrape data seamlessly from e-commerce websites like Walmart. These can include anti-bot policies, JavaScript protection, dynamic HTML structures, IP blocking and more. Let’s look into some of these challenges in a little more detail.

**Challenges of Web Scraping with ChatGPT**

While ChatGPT serves as a powerful tool for web scraping simple websites, when it comes to scraping data from the websites of e-commerce giants like Walmart, it runs into several challenges. These include:

* **JavaScript Rendering**: Modern websites rely on JavaScript to load content dynamically, which makes them way faster. Examples are infinite scroll websites, pop-up windows and interactive widgets.
* **Rate Limit and IP blocking:**Excessive requests from a single IP address can trigger web scraping blocks. Rotating IP addresses helps prevent this.
* **Changes In Architecture:**Websites change their HTML architecture from time to time. This means new prompts need to be constantly created to match the changes.
* **ReCAPTCHAS and anti-robots:**Some websites use several anti-bot features such as reCAPTCHAS to avoid huge traffic from web scraping.
* **Outdated Information:**The data used to train GPT3.5 and GPT4, may be outdated when it comes to web scraping libraries.

Getting around these challenges manually is not easy and can lead to increasing code complexity and reliance on third-party libraries which come with their own infrastructural costs.

This is where the [Nimble Browser](https://nimbleway.com/nimble-api/browser/) stands as a robust solution to the challenges mentioned above, offering cutting-edge residential proxies, LLM technology, and powerful unblocking capabilities.

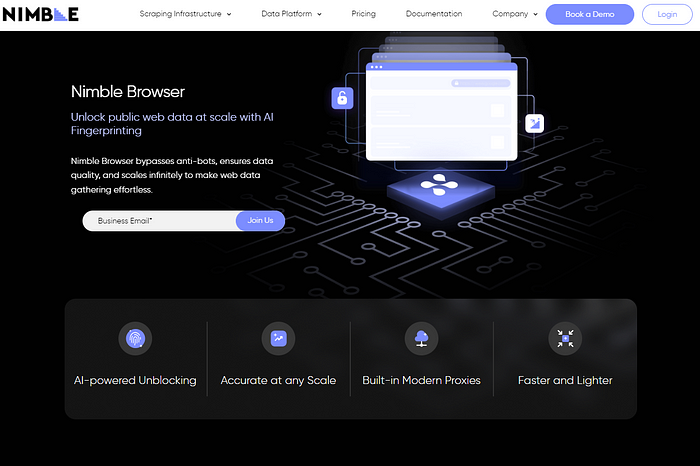


Image from [Nimble’s website](https://nimbleway.com/nimble-api/browser/" \t "_blank)

**[Nimble Browser | A Browser Designed for Web Scraping - Nimble](https://tracking.nimbleway.com/SH20?source=post_page-----29ce04818e72--------------------------------" \t "_blank)**

[Nimble Browser is a lightweight automated browser designed specifically for effortless web data scraping and…](https://tracking.nimbleway.com/SH20?source=post_page-----29ce04818e72--------------------------------" \t "_blank)

[nimbleway.com](https://tracking.nimbleway.com/SH20?source=post_page-----29ce04818e72--------------------------------" \t "_blank)

The Nimble Browser is a comprehensive solution that includes all of Nimble’s advanced technologies in one, such as its various Web APIs and premium proxy solutions for a seamless web scraping experience. Using the Nimble Browser, you can easily overcome the challenges mentioned above. It includes:

* Nimble’s various advanced [web scraping platform APIs](https://tracking.nimbleway.com/SH21) — such as the Web API, E-commerce API, SERP API, Maps API — that are fine-tuned for specific use cases. Nimble’s Web APIs utilize state-of-the-art AI-powered parsing and structuring for faster and more accurate results than traditional CSS selector or xPath-based scrapers, allowing for data extraction at any granularity you’d like.
* Nimble’s [premium zero-downtime IP infrastructure](https://tracking.nimbleway.com/SH1z) which includes its vast network of [residential proxies](https://tracking.nimbleway.com/SH1z), easily letting you bypass IP-based rate limits and reCAPTCHAs, while ensuring maximum success rates even for particularly sensitive pages. Nimble’s IP infrastructure is AI-optimized to intelligently select the best premium IPs for every request, and you can even granularly target down to state and city levels for any geolocation in the world, making localized web data gathering a breeze.
* One of the major challenges faced while scraping the web is bypassing device/browser fingerprints which flags and blocks your request as coming from a bot or crawler. Nimble’s[fingerprint engine](https://docs.nimbleway.com/nimble-api/nimble-api-quick-start-guide" \l "whats-included-with-nimble-api" \t "_blank) automatically manages TLS fingerprints, canvas checking, headers, cookies, and more, ensuring unrestricted and uninterrupted access to public web data.
* Page interactions which enable users to [perform operations on a webpage before the web data is collected and returned](https://docs.nimbleway.com/nimble-api/web-api/page-interactions), such as clicking, typing, and scrolling. These are synchronous and run sequentially one by one, with an overall sequence limit of 60 seconds. They are supported by real-time, asynchronous, and batch requests. This feature is particularly helpful for pages that are dynamic or using lazy loading and infinite scrolling.

[***👉 Explore and learn more about the Nimble Browser and its functionalities***](https://tracking.nimbleway.com/SH20)

**Scraping Walmart Using ChatGPT: A Step-By-Step Guide**

We’re now going to scrape Walmart using ChatGPT. More precisely, we’ll crawl information about mirrorless cameras from the following page:

**[Mirrorless Cameras in Cameras & Camcorders - Walmart.com](https://www.walmart.com/browse/electronics/mirrorless-cameras/3944_133277_1230677?source=post_page-----29ce04818e72--------------------------------" \t "_blank)**

[Shop for Mirrorless Cameras in Cameras & Camcorders. Buy products such as Canon EOS R8 Mirrorless Camera with 24-50mm…](https://www.walmart.com/browse/electronics/mirrorless-cameras/3944_133277_1230677?source=post_page-----29ce04818e72--------------------------------" \t "_blank)

[www.walmart.com](https://www.walmart.com/browse/electronics/mirrorless-cameras/3944_133277_1230677?source=post_page-----29ce04818e72--------------------------------" \t "_blank)

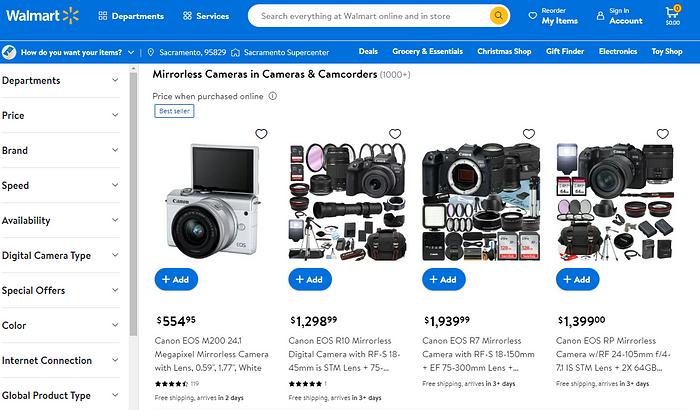


Image from [Walmart’s website](https://www.walmart.com/browse/electronics/mirrorless-cameras/3944_133277_1230677)

The target [Walmart’s page](https://www.walmart.com/browse/electronics/mirrorless-cameras/3944_133277_1230677) contains details about mirrorless cameras, such as the price in USD, the product’s title, the rating and the number of reviews.

**The Steps**

First, you need to create a ChatGPT account, if you don’t have one already. Go to the [login page](https://chat.openai.com/auth/login) and sign up. Once in, we need to instantiate the first prompt by providing the URL to ChatGPT and specify which libraries we want to use. Let’s use [Beautiful Soup](https://www.crummy.com/software/BeautifulSoup/bs4/doc/), which is the most widely used HTML parser, hence it must be a well-known library for the GPT3.5 model. And since we’ll be using Python, we also need to use the [requests](https://pypi.org/project/requests/) package.

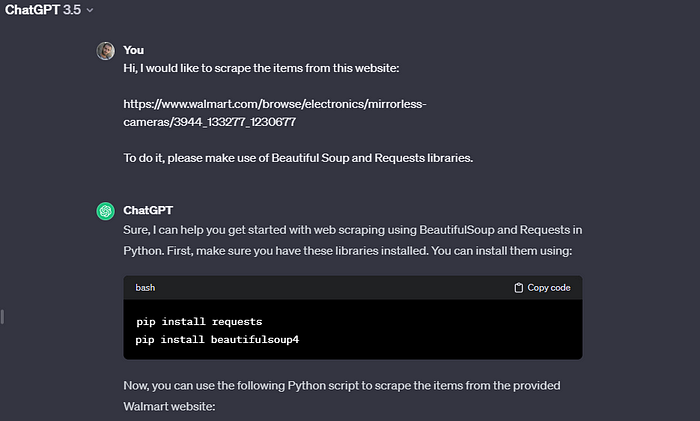


Image from [ChatGPT](https://chat.openai.com/)

ChatGPT starts by providing the pip commands to install the previously mentioned libraries. Then it delves into the actual code structure:

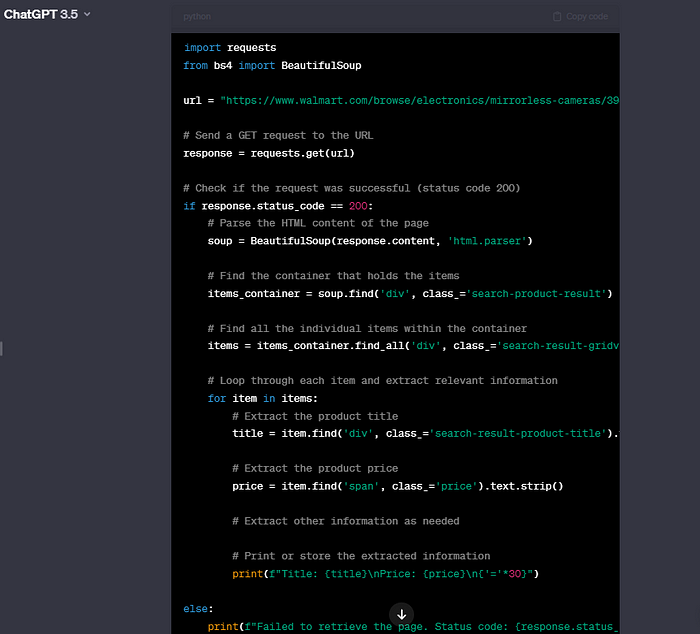
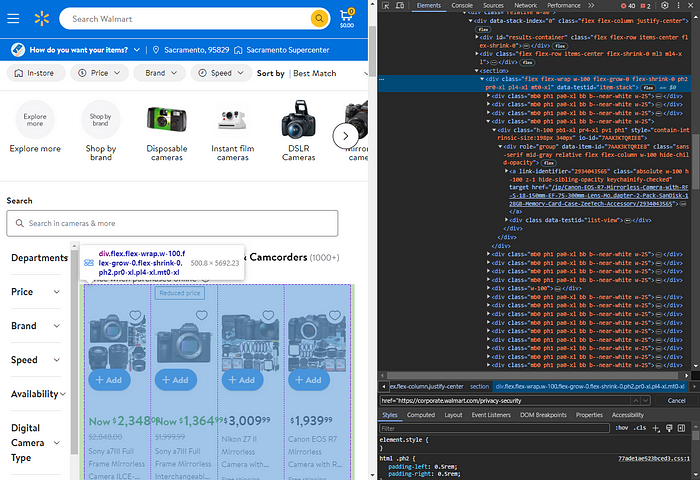


Image from [ChatGPT](https://chat.openai.com/)

ChatGPT does not know the HTML elements classes we want to scrape but provides a backbone, which we can use to make modifications. For instance, for the items container, we can replace the class search-result-gridview-item with the actual class. To obtain it, navigate to the desired web page, right-click, and select Inspect*.*The HTML code and the associated classes for each element will be displayed on the right side.



Inspect [Walmart’s page](https://www.walmart.com/browse/electronics/mirrorless-cameras/3944_133277_1230677)

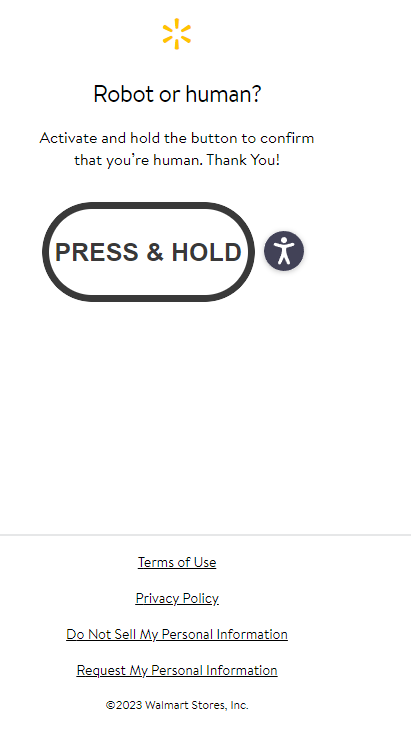
Now the items\_container object uses the class selected in the image above. The updated code should look like this:

import requests  
from bs4 import BeautifulSoup  
  
url = "https://www.walmart.com/browse/electronics/mirrorless-cameras/3944\_133277\_1230677"  
  
# Send a GET request to the URL  
response = requests.get(url)  
  
# Check if the request was successful (status code 200)  
if response.status\_code == 200:  
 # Parse the HTML content of the page  
 soup = BeautifulSoup(response.content, 'html.parser')  
  
 # Find the container that holds the items  
 items\_container = soup.find(  
 'div',  
 class\_='flex flex-wrap w-100 flex-grow-0 flex-shrink-0 ph2 pr0-xl pl4-xl mt0-xl')  
  
 # Find all the individual items within the container  
 items = items\_container.find\_all(  
 'div', class\_='mb0 ph1 pa0-xl bb b--near-white w-25')  
  
 # Loop through each item and extract relevant information  
 for item in items:  
 # Extract the product title  
 title = item.find(  
 'span', class\_='w\_V\_DM').text.strip()  
  
 # Extract the product price  
 price = item.find('span', class\_='w\_iUH7').text.strip()  
  
 # Extract other information as needed  
  
 # Print or store the extracted information  
 print(f"Title: {title}\nPrice: {price}")  
  
else:  
 print(f"Failed to retrieve the page. Status code: {response.status\_code}")

But when you run the script above in your Python IDE, you’ll notice an error in the output:

AttributeError: 'NoneType' object has no attribute 'find\_all

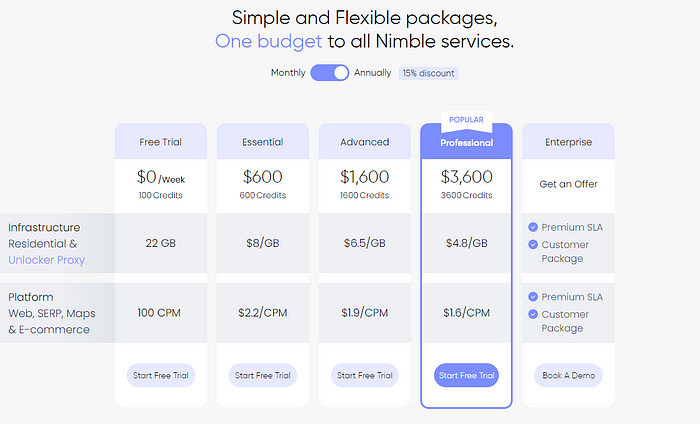
This happens because the URL we are scraping is protected by a reCAPTCHA, where the classes we are using do not exist.



Walmart’s reCAPTCHA

At this stage, we could ask ChatGPT to use a headless browser like [Selenium](https://www.selenium.dev/) or [Playwright](https://playwright.dev/python/), to solve the reCAPTCHA, but this would be a time-consuming task and with some degree of difficulty. To ease the process, we can jump directly to the [Nimble Browser’s](https://tracking.nimbleway.com/SH20) API solutions, in this case, the [E-commerce API](https://nimbleway.com/nimble-api/ecommerce/).

To create an account on Nimble, you can simply do it by selecting a [plan of your choice](https://tracking.nimbleway.com/SH2N) (for example, you can select **‘Start Free Trial’** under **‘Free Trial’**to opt for a free trial and check all their products out), and then filling in your details.



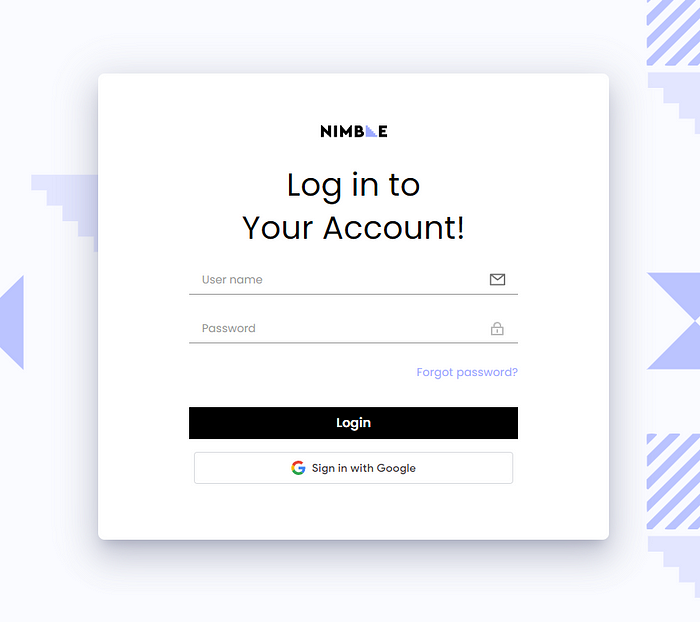
Nimble’s [pricing options](https://nimbleway.com/pricing/?aff_id=48&offer_id=4&transaction_id=102b02678b37de5c16c2166c8b85e1)

**[Buy Proxies and Web Scraping APIs - Nimble](https://tracking.nimbleway.com/SH2N?source=post_page-----29ce04818e72--------------------------------" \t "_blank)**

[Looking to buy proxies or advanced web scraping tools? Check out Nimble's flexible pricing packages. Start with a free…](https://tracking.nimbleway.com/SH2N?source=post_page-----29ce04818e72--------------------------------" \t "_blank)

[nimbleway.com](https://tracking.nimbleway.com/SH2N?source=post_page-----29ce04818e72--------------------------------" \t "_blank)

And then you can log in with your Google account [here](https://app.nimbleway.com/login).



Nimble’s [login page](https://app.nimbleway.com/login)

After completing the registration process, we can initiate the integration of Nimble’s E-commerce API into a Python script.

import requests  
import base64  
  
username = '<your\_nimble\_username'  
password = '<your\_nimble\_password'  
token = base64.b64encode(bytes(f"{username}:{password}", 'utf-8'))  
token = token.decode('utf-8') # convert bytes to string

In the script above, we start by importing the necessary libraries. The[requests](https://pypi.org/project/requests/) library will serve to make an HTML request to the Nimble’s API and the [base64](https://docs.python.org/3/library/base64.html) provides functions for encoding binary data to printable ASCII characters. The latter is important because Nimble APIs use [Basic access authentication](https://en.wikipedia.org/wiki/Basic_access_authentication) with a credential string (token) generated from the username and password.

base64(username:password)

In Python, the encoding is achieved with the function base64.b64encode() and the strings username and password are converted to bytes with the encoding utf-8. This output is then decoded to a string which will be the token used with the Basic access authentication.

Let’s now define the API URL and Walmart’s page containing the mirrorless cameras.

api\_url = 'https://api.webit.live/api/v1/realtime/ecommerce'  
url = 'https://www.walmart.com/browse/electronics/mirrorless-cameras/3944\_133277\_1230677'  
  
headers = {  
 'Authorization': f"Basic {token}",  
 'Content-Type': 'application/json'  
}  
  
data = {  
 "parse": True,  
 "vendor": "walmart",  
 "url": url,  
 "format": "json",  
 "country": 'us',  
 "locale": "en",  
}  
  
response = requests.post(api\_url, headers=headers, json=data)

At the beginning of the script, we see the E-Commerce API URL (api\_url) and the URL to scrape. The headers dictionary contains the Content-Type and the Authorization, which takes the token previously generated with base64.

The data dictionary has several keys to define the scraping process. We set parse to True at first, then the vendor is *walmart,*but others can be used by Nimble’s E-Commerce API. When it comes to the format we can select different types, such as list, table, object and json. In this case, we use json to facilitate the data processing step.

As for the country key, as mentioned before, Nimble uses [residential proxies](https://tracking.nimbleway.com/SH1z) which allows the user to scrape from different geolocations. The locale is the language at the output.

Finally, we use the requests instance, to make a POST HTML request with the provided API URL and the dictionaries.

We’re now ready to run the script, we just need to save the .json file somewhere in our directory. For that, we can use the following:

import json  
  
with open('data.json', 'w', encoding='utf-8') as f:  
 json.dump(response.json(), f, ensure\_ascii=False, indent=4)

Once the script is launched, we’ll get a data.json filled with all the items captured by the API. To easily navigate through the file, I recommend using a JSON extension. Let’s now see the results for the first item:

{  
 '\_\_typename': 'Product',  
 'additionalOfferCount': None,  
 'annualEvent': False,  
 'annualEventV2': False,  
 'arExperiences': {'isARHome': False, 'isAROptical': False, 'isZeekit': False},  
 'aspectInfo': {'header': None, 'id': None, 'name': None, 'snippet': None},  
 'availabilityInNearbyStore': None,  
 'availabilityStatusDisplayValue': 'In stock',  
 'availabilityStatusV2': {'display': 'In stock', 'value': 'IN\_STOCK'},  
 'averageRating': 4.8,  
 'badge': {'bundleId': '', 'id': '', 'key': '', 'text': '', 'type': ''},  
 'badges': {  
 'flags': None,  
 'groups': [  
 {  
 '\_\_typename': 'UnifiedBadgeGroup',  
 'members': [  
 {  
 '\_\_typename': 'BadgeGroupMember',  
 'badgeContent': None,  
 'id': 'L1053',  
 'key': 'FF\_SHIPPING',  
 'memberType': 'badge',  
 'otherInfo': None,  
 'rank': 1,  
 'slaText': 'in 2 days',  
 'styleId': 'FF\_STYLE',  
 'templates': None,  
 'text': 'Free shipping, arrives ',  
 'type': 'LABEL'  
 }  
 ],  
 'name': 'fulfillment'  
 }  
 ],  
 'tags': []  
 },  
 'brand': None,  
 'buyBoxSuppression': False,  
 'canAddToCart': True,  
 'canonicalUrl':   
'/ip/Sony-a7III-Full-Frame-Mirrorless-Camera-ILCE-7M3K-B-2-Lens-SEL2870-FE-28-70mm-F3-5-5-6-OSS-SEL85F18-85mm-F1-8-Set-Deco-Gear-Case-x-64GB-Memory-Cards/329661935',  
 'category': {'path': None},  
 'checkStoreAvailabilityATC': False,  
 'classType': 'REGULAR',  
 'description': '24.2MP 35mm full-frame back-illuminated CMOS sensorSensitivity range up to ISO 51200Fast Hybrid ...',  
 'earlyAccessEvent': False,  
 'entity\_type': 'SearchResult',  
 'esrb': None,  
 'eventAttributes': {'priceFlip': False, 'specialBuy': False},  
 'externalInfo': None,  
 'externalInfoUrl': '',  
 'fitmentLabel': None,  
 'flag': '',  
 'fulfillmentBadgeGroups': [{'className': 'dark-gray', 'isSlaTextBold': True, 'slaText': 'in 2 days', 'templates': None, 'text': 'Free shipping, arrives '}],  
 'fulfillmentBadges': [],  
 'fulfillmentSpeed': ['2 days'],  
 'fulfillmentSummary': [{'deliveryDate': None, 'storeId': '0'}],  
 'fulfillmentTitle': 'title\_shipToHome\_not\_available',  
 'fulfillmentType': 'MARKETPLACE',  
 'groupMetaData': {'groupComponents': None, 'groupSubType': None, 'groupType': None, 'numberOfComponents': 0},  
 'hasCarePlans': True,  
 'hasSellerBadge': None,  
 'id': '7EW9G9KSS1MV',  
 'image': 'https://i5.walmartimages.com/asr/a5a343b9-a588-431d-96d3-672a780c08b1\_1.54d75600bbd76a9f06f9cef9b8df6279.jpeg?odnHeight=180&odnWidth=180&odnBg=ffffff',  
 'imageID': 'BFAAC2D5FDF448DE80AC0AF4955BEEEA',  
 'imageInfo': {  
 'id': 'BFAAC2D5FDF448DE80AC0AF4955BEEEA',  
 'name': 'a5a343b9-a588-431d-96d3-672a780c08b1\_1.54d75600bbd76a9f06f9cef9b8df6279.jpeg',  
 'size': '290-392',  
 'thumbnailUrl':   
'https://i5.walmartimages.com/asr/a5a343b9-a588-431d-96d3-672a780c08b1\_1.54d75600bbd76a9f06f9cef9b8df6279.jpeg?odnHeight=180&odnWidth=180&odnBg=ffffff'  
 },  
 'imageName': 'a5a343b9-a588-431d-96d3-672a780c08b1\_1.54d75600bbd76a9f06f9cef9b8df6279.jpeg',  
 'imageSize': '290-392',  
 'isAtfMarker': False,  
 'isEarlyAccessItem': False,  
 'isOutOfStock': False,  
 'isPreowned': False,  
 'isSponsoredFlag': False,  
 'isVariantTypeSwatch': False,  
 'itemType': None,  
 'keyAttributes': [],  
 'manufacturerName': None,  
 'mediaRating': None,  
 'moqText': None,  
 'name': 'Sony a7III Full Frame Mirrorless Camera ILCE-7M3K/B with 2 Lens SEL2870 FE 28-70mm F3.5-5.6 OSS and SEL85F18 FE 85mm F1.8 Set + Deco Gear Case 2 x 64GB   
Memory Cards Extra Battery Kit Deluxe Bundle',  
 'newConditionProductId': None,  
 'numberOfReviews': 20,  
 'offerId': '48C2568E88F54B86BA4CE265781D175E',  
 'pac': None,  
 'petRx': {'eligible': False, 'singleDispense': None},  
 'pglsCondition': None,  
 'preOrder': {  
 'isPreOrder': False,  
 'preOrderMessage': None,  
 'preOrderStreetDateMessage': None,  
 'streetDate': None,  
 'streetDateDisplayable': None,  
 'streetDateType': None  
 },  
 'preownedCondition': 'New',  
 'price': 2348,  
 'priceFlip': False,  
 'priceInfo': {  
 'eaPricingPreText': '',  
 'eaPricingText': '',  
 'finalCostByWeight': False,  
 'itemPrice': '$2,848.00',  
 'linePrice': '$2,348.00',  
 'linePriceDisplay': 'Now $2,348.00',  
 'memberPriceString': '',  
 'minPrice': 0,  
 'minPriceForVariant': '',  
 'priceDisplayCondition': '',  
 'priceRangeString': '',  
 'savings': 'SAVE $500.00',  
 'savingsAmt': 500,  
 'shipPrice': '',  
 'submapType': '',  
 'subscriptionDualPrice': None,  
 'subscriptionPercentage': None,  
 'subscriptionPrice': '',  
 'subscriptionString': '',  
 'unitPrice': '',  
 'wasPrice': '$2,848.00'  
 },  
 'productLocation': None,  
 'productLocationDisplayValue': None,  
 'promoDiscount': None,  
 'quickShop': None,  
 'quickShopCTALabel': None,  
 'rating': {'averageRating': 4.8, 'numberOfReviews': 20},  
 'rewards': None,  
 'salesUnit': 'EACH',  
 'salesUnitType': 'EACH',  
 'seeShippingEligibility': False,  
 'sellerId': '10D2E6BEB0F84585BF5E7BFF5332EE60',  
 'sellerName': 'Beach Camera ',  
 'shortDescription': '24.2MP 35mm full-frame back-illuminated CMOS sensorSensitivity range up to ISO 51200Fast Hybrid ...',  
 'shouldLazyLoad': False,  
 'showAtc': True,  
 'showBuyNow': False,  
 'showExploreOtherConditionsCTA': False,  
 'showOptions': False,  
 'similarItems': False,  
 'snapEligible': False,  
 'specialBuy': False,  
 'sponsoredProduct': None,  
 'subscription': {'\_\_typename': 'SubscriptionData', 'subscriptionEligible': False},  
 'topResult': None,  
 'type': 'REGULAR',  
 'usItemId': '329661935',  
 'variantCriteria': [],  
 'variantList': [],  
 'vision': {'ageGroup': None, 'visionCenterApproved': False},  
 'weightIncrement': 1  
}

The output above, not only gives the information that we were trying to obtain using ChatGPT, but many many more additional insights that would be worth looking into. The output is obtained in less than three seconds and there’s no need to write prompts or tweak them continuously to achieve the scraping goals.

**Conclusion**

This article explored ChatGPT as a web scraping solution, suitable for basic websites that lack sophisticated anti-bot mechanisms. While incorporating headless browser libraries through ChatGPT prompts might circumvent certain blocking measures, their effectiveness is not guaranteed.

With [Nimble’s solutions](https://tracking.nimbleway.com/SH20" \t "_blank), the scraping is done at extremely high speed and without blocking, which makes it a preferred tool when it comes to data collection at scale. Its extensive list of residential proxies allows you to scrape without limitations from the seven corners of the world. No IP blocking and no rate limitation. The output is delivered in a surprisingly structured way, which facilitates easy integration with any infrastructure.

I would recommend taking the [free trial](https://tracking.nimbleway.com/SH2N) and giving Nimble a try for yourself.

[Web Scraping](https://medium.com/tag/web-scraping?source=post_page-----29ce04818e72---------------web_scraping-----------------)

[ChatGPT](https://medium.com/tag/chatgpt?source=post_page-----29ce04818e72---------------chatgpt-----------------)

[Ecommerce](https://medium.com/tag/ecommerce?source=post_page-----29ce04818e72---------------ecommerce-----------------)

[API](https://medium.com/tag/api?source=post_page-----29ce04818e72---------------api-----------------)

[Python](https://medium.com/tag/python?source=post_page-----29ce04818e72---------------python-----------------)