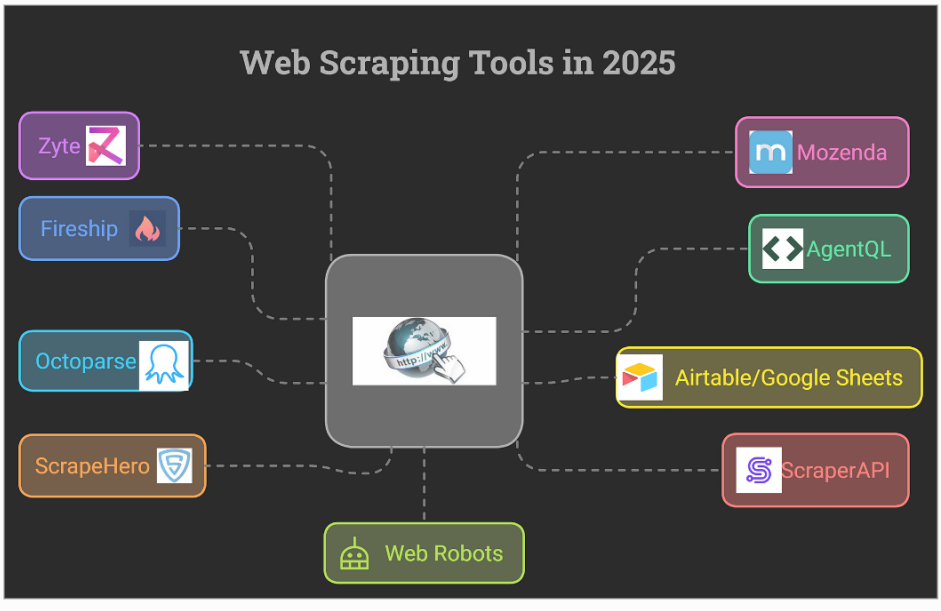
Use AI to Scrape Almost All Websites Easily in 2025

Manpreet Singh
.
Follow
Published in
AI Advances
7 min read
.
Oct 30, 2024
Listen
Share
More



Created by author

Hi everyone!

Today, I'm going to show you an easy way to collect data from lots of websites and how to build a web scraper that can work just like a human using a browser.

This kind of scraper can even do freelance web scraping jobs on its own, on sites like Upwork.

Web scraping has changed a lot because of AI, especially in 2024.

In the past, big companies like Amazon or Walmart had to spend a lot of time and money to scrape data from other websites so they could keep their prices competitive.

They did this by copying what a browser does: sending requests to get the website's HTML, and then using special code to find and get the information they wanted.

This was hard because each website is different, and if a website changed its design, the scraper stopped working. This meant companies had to spend more time fixing and updating their scrapers.

Imagine Amazon wants to keep an eye on what Walmart is charging for the same products. To do this, Amazon would need a scraper made just for Walmart's site.

But if Walmart changed their site, Amazon would have to update the scraper.

This is time-consuming and costly.

It's not just big companies that need scrapers.

If you check freelance websites like Upwork, you'll see lots of small businesses looking for people to build scrapers for things like finding contact details, tracking prices, market research, or job listings.

For example, a small startup may need to monitor product prices across different e-commerce websites to set their own prices.

Before AI, it was hard and expensive for small businesses to get these solutions.

Now, with the help of large language models (LLMs) and new tools, it's much easier and cheaper to create web scrapers. What used to take a week for a developer to build can now be done in just a few hours. LLMs can understand different website structures in a smarter way, so you don't need to keep rewriting scrapers for every small change.

Let's talk about how to scrape data effectively and handle different kinds of websites — from simple ones to really complicated ones.

I'll break this down into three groups:

- simple public websites
- websites that need more complex workflows
- advanced cases that need smart agents.

1. Scraping Simple Public Websites

wikipedia.org

Simple public websites are pages like Wikipedia or company websites that don't need you to log in or pay.

These sites can still be challenging because they have different layouts, but with large language models, this job has become much easier.

Let's say you need to collect information about different plants from Wikipedia for a school project.

In the past, you would need to look at the HTML code of each page, find the tags with the data you need, and then write custom code to get that data.

Doing this for every page would be a lot of work.

But now, with LLMs, you can just give the raw HTML to the AI and it can extract the data for you.

- 

view-source- wikipedia.org

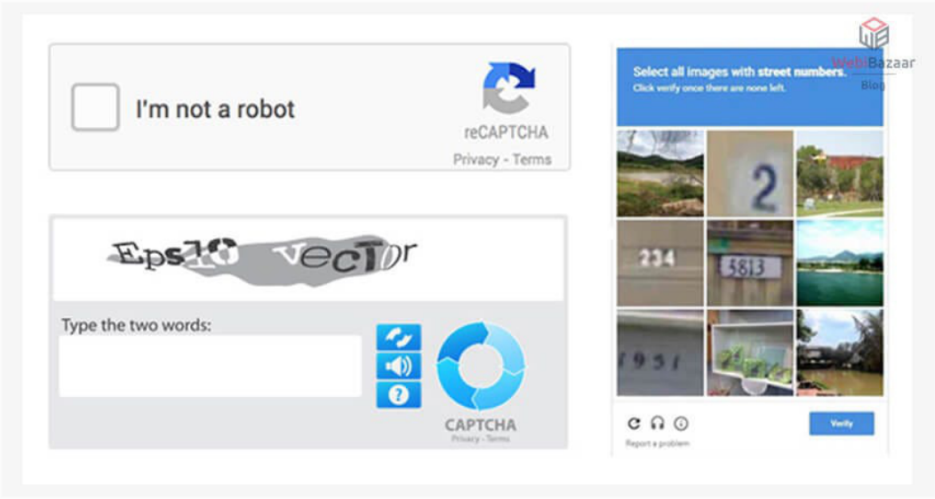
You can even tell it exactly what data you need, like "get the plant's name, description, and care tips," and the AI will give you a nicely organized answer.

This saves a lot of time and effort.

LLMs are also good at figuring out where information is if you don't know the exact page it's on.

For example, if you are looking for contact information on a company website but aren't sure which page has it, the AI-powered scraper can search all the pages until it finds what you need. It's like having a helper that knows where to click and what to read.

2. Scraping Websites with Complex Interactions

- 

webibazaar.com

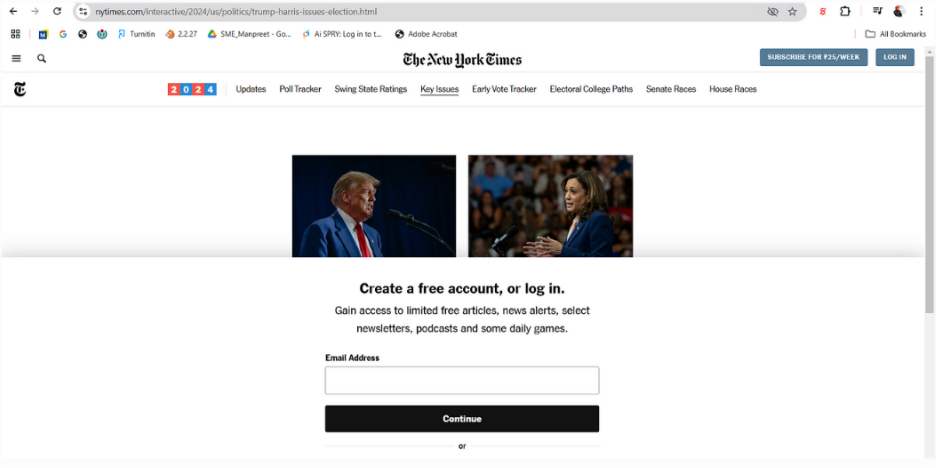
Some websites are harder to scrape because they need you to interact with them — like logging in, solving CAPTCHAs, or clicking past pop-ups.

Think of news websites that need you to log in to see the articles. This is where tools like

- Selenium
- Puppeteer
- Playwright help

These tools were originally made for testing websites, but now they are used to simulate how a real person would use a website.

Imagine you want to scrape articles from a news site like The New York Times. The articles are behind a paywall, so you need to log in first.

- 

You can use tools like **Playwright or Selenium** to make the scraper log in for you, click through pop-ups, and access the articles.

But even with these tools, it can still be tricky to make the scraper interact with every button or input box on the page.

This is where AgentQL comes in handy.

AgentQL helps find the right elements on a webpage, like buttons and forms, and tells the scraper what to do.

For example, if you want to collect job listings from multiple job boards, AgentQL can help your scraper find the login form, fill it in, and navigate to the job postings.

This means you can collect a lot of job listings in just a few minutes without doing any manual work.

You could even have the scraper put the data into Google Sheets or Airtable, so it's easy to sort and analyze.

Let's say you are trying to keep track of software developer job openings on sites like **Indeed, Glassdoor, and LinkedIn.**

With these tools, you can make the scraper log in, search for jobs, and gather all the details in one place, like a Google Sheet.

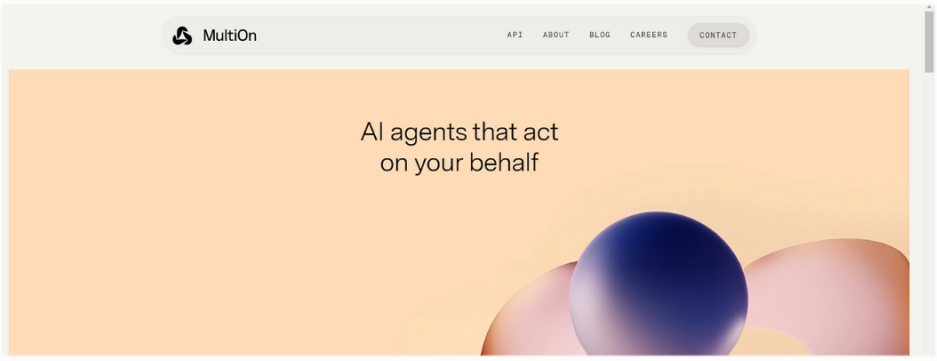
This saves you hours of work.

3. Advanced Uses That Need Smart Thinking

The last group involves more vague tasks that need decision-making — like finding the ***cheapest flight to a destination within the next two months or buying a concert ticket based on your budget***.

These tasks are tough because they require planning and judgment. While still new, there are tools being developed that can do this.

One such platform is Multion, which makes agents that can do these kinds of complex tasks on their own.

- 

multion.ai

For example, you could ask the agent to

“find and book the cheapest flight from New York to Melbourne in July,”

and it would look through different travel websites, compare prices, and book the flight for you.

It’s not perfect yet, but it’s impressive how close these tools are to acting like a real person.

Another example is buying a concert ticket. You could ask an agent

“Buy me a ticket for Taylor Swift’s concert for under $100.”

The agent would look through different ticket sites, find a ticket that meets your budget, and make the purchase.

This technology is still developing, but tools like Multion are making it possible to automate even these tricky tasks.

Practical Tools for Web Scraping

Here are some useful tools if you want to start web scraping using LLMs and agents:

- **Fireship, Gina, and SpiderCloud**: These tools help turn web content into an easy-to-read format that AI models can understand better. For example, Fireship can take a complicated restaurant website and turn it into a simple version that only includes the important information, like menu items and prices. This makes it cheaper and faster for AI models to process the information.
- **AgentQL**: This tool helps a scraper interact with websites just like a person would. For example, if you need to scrape a job board that has lots of buttons to click and forms to fill out, AgentQL helps make sure your scraper can do all of that smoothly.
- **Airtable/Google Sheets Integration**: Once your scraper collects data, it’s important to save it in a useful way. Tools like Airtable or Google Sheets can store the data so you can easily analyze it later. For example, if you are tracking house prices on real estate websites, Google Sheets can help you compare and analyze trends over time.
- **Octoparse and ScrapeHero**: These tools are really good at handling JavaScript-heavy websites. Octoparse has prebuilt templates that make it easy to scrape data from e-commerce websites, and it uses smart methods to avoid getting blocked. ScrapeHero is great for projects that need a lot of data quickly, like collecting prices from many stores at once.
- **ScraperAPI and Zyte**: These services help make sure your scraper doesn’t get blocked by rotating proxies. ScraperAPI lets you customize things like request headers, which is useful for targeted scraping. Zyte, formerly called Scrapy, is also really good at handling big scraping jobs and making sure you get the data you need without being interrupted.
- **Mozenda and Web Robots**: Mozenda helps automate more complicated web forms and also allows you to schedule scraping tasks. Web Robots is great if you need to create your own scraping programs and extract data directly into files like CSV or Excel.

So, in 2024 and 2025, AI is changing the way we scrape data from websites.

With large language models and tools like AgentQL and Playwright, even complex sites can be scraped with less manual work.

The best part is that these systems are flexible enough to handle a wide variety of tasks — whether it’s collecting business data, searching for jobs, or even booking flights.

The opportunities to automate web scraping are bigger and more accessible than ever.

So, whether you’re a small business wanting market data, a freelancer helping a client, or someone just curious to learn ? then these AI tools make web scraping a powerful and simple solution.

Must try !!!

Technology

Artificial Intelligence

Data Science

Programming

Web3

556

6

- 
- 

Follow

Written by Manpreet Singh

272 Followers

·Writer for

AI Advances

Data Scientist | AI | Machine Learning |Research & Technical Writer Connect with me on LinkedIn to collaborate : https://www.linkedin.com/in/manpreet17/

-