Back to Blog

Web Scraping Best Practices That Keep Everyone Happy

Alli Davis
October 19, 2021

Community

As you may already know, pulling data from the web at a large scale is an excellent way to gather valuable information that can help you improve business operations and increase your revenue. However, most sites frown upon this data collection technique since they have a hard time distinguishing legitimate researchers from malicious actors. Moreover, their businesses could be harmed by greedy bot activity overwhelming their servers. That's why many of them have harsh anti-bot measures in place.

Table of Contents

In theory, you could perform your data-gathering tasks manually. However, not only do most business owners not have the time for this, but it's also not a practical solution. Your best bet is to get a bot to do the leg work for you. But you need to understand web scraping best practices to avoid getting in trouble while automating your research.

This guide will tell you all the web scraping guidelines you need to know to succeed in your data-gathering endeavors. Feel free to use the table of contents below to skip to the parts that interest you the most.

Common Challenges When Web Scraping

Even veteran web scrapers can face a number of problems while trying to extract the data they need from certain sites. Have a look at the most common pitfalls you could potentially experience while web scraping, and the best practices in scraping data from the web.

1. Messy website structure or changes to the HTML

Sometimes, the root of your web scraping problems is not anti-scraping measures on the sites you're trying to scrape. It could be that what's causing errors in your script is a differing layout between pages of a website, or that your web scraper is encountering unstructured datasets. Unless you use a system that reports all

2. Accidentally scraping the wrong data

If you're just scraping a few pages, you might be in the clear here. However, if you're doing high-volume scraping, it's easy to lose track of the data you've already gathered and end up with duplicate data, or the wrong data altogether.

Make sure you program your bot so that the data scraped meets your quality guidelines. Also, check for sites that use different URLs to direct users to the same information. Using the right software can detect and prevent duplicate values.

3. CAPTCHAs and IP bans

The term CAPTCHA stands for Completely Automated Public Turing test to tell Computers and Humans Apart. Even if you don't frequently browse the web, you've probably stumbled upon one of these bot-detecting puzzles at least once. They typically require you to identify a series of images, retype a distorted sequence of letters and numbers, or simply check a box to prove you're human. If you fail, you're simply not allowed to access the content you're looking for.

Another common anti-scraping measure is IP tracking and blocking. Some sites have adopted IP fingerprinting to block and ban bots. They generally keep a record of the IP addresses used to send requests to their servers and other browser-related parameters. If they suspect a specific IP is attached to a robot, they might block it from entering the site. Blocks are usually temporary unless more serious rules have been violated.

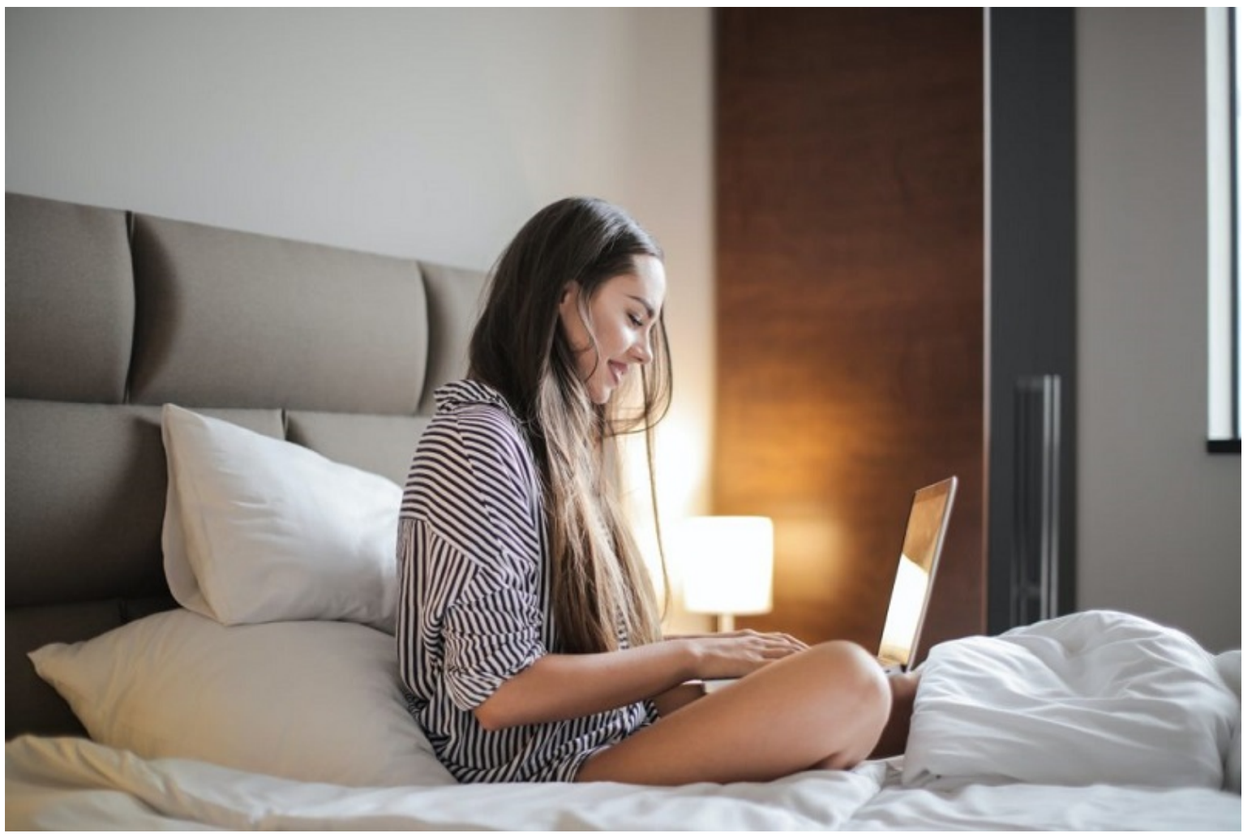
4. AJAX elements

Some sites use AJAX or Asynchronous JavaScript and XML to create websites that do not require a page refresh in order to load data from the server. This type of programming is used to create pages with infinite scroll. Sites that use this JavaScript technology are quite a challenge to scrape because they show data after the HTML has already loaded. Scrapers need a way to execute and render JavaScript in order to extract data from these sites.

5. Honeypot traps

Some websites have more clever techniques to keep web scrapers at bay. One of them is implementing honeypot traps, which are invisible links that only bots can find and click on. These links are typically hidden behind CSS attributes or camouflaged with the background color of the page. Once a bot finds and clicks on them, it's automatically labeled and blocked by the site.

Web Scraping Best Practices



The main reason most websites have strict anti-scraping measures in place is due to those who have malicious intent for the data they collect, such as an intent to

Help

quickly. You have to keep in mind that you're pretty much asking for a huge favor when extracting data from a particular page, and you need to be a good guest in return.

Abiding by the rules is a must if you don't want sites to end up implementing more stringent anti-bot techniques that will make your job much harder. Keep it ethical, and the rest should come easy. Here are the best practices for web scraping.

1. Respect robots.txt file

Most sites have specific rules for good scraping behavior. Said regulations typically appear on the website's robots.txt file, and include specifics on how frequently you can send requests, which pages you're allowed to extract data from, etc. In some cases, this file will even dictate whether or not you're allowed to scrape at all. If the robot.txt file of a particular site says not to, you'd better stop. In all cases, be respectful of the boundaries a site has in place.

2. Slow down your requests

A common giveaway for scraping bots is how fast they submit their request to the server, as they can comb websites far more quickly than humans can. What's more, too many requests sent too quickly could easily overwhelm the system and make the site crash, affecting the site's user experience and potentially making the site owners lose clients and revenue.

You should always space out your requests by at least 10 seconds — or even more during peak hours. Add some delays and programmatic sleep calls to your script to make it look like it's a regular user, and not a robot, who's sending the requests.

3. Switch crawling patterns

Humans are unpredictable creatures. We tend not to perform repetitive tasks as we browse through a particular site — or at least not as precisely as a robot would. We typically include random actions here and there, and that's the behavior your scraping bot should mimic. Incorporate aleatory mouse movements and other actions that will prevent the anti-crawling mechanisms from getting triggered.

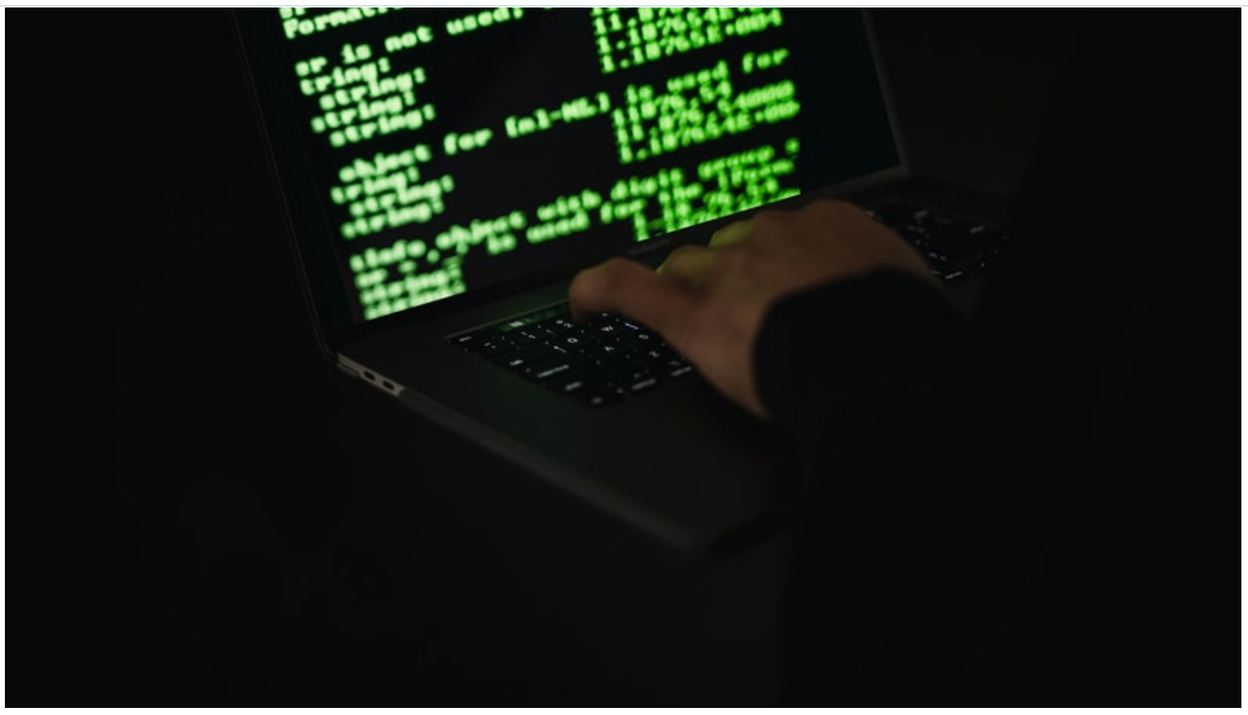
4. Don't violate copyright

By definition, copyright is the exclusive legal ownership over an original and tangible piece of work. This means others cannot use it without the explicit authorization of the owner. It's common to stumble upon copyrighted content while testing your web scraping techniques, especially when extracting data from:

- Videos
- Images
- Articles
- Music

To ensure you don't run into any web scraping copyright issues, always respect the fair use exceptions.

How can websites detect and block web scraping?



You might be wondering, how can websites detect suspicious web scraping activity? There are numerous mechanisms to detect crawlers and spiders and stop them in their tracks. These methods include:

- **Monitoring traffic and download rate** — sites use your IP address to keep tabs on how many requests you send within a certain period. That's why it's important to use proxies.
- **Repetitive tasks** — as mentioned above, humans tend not to follow the same patterns while browsing the web. Monotonous and predictable patterns are a clear giveaway that a bot is sending the requests to the server.
- **Hidden links** — If your bot is not programmed to detect hidden links on a site's CSS code, it may trigger a honeypot trap. These are unlikely to be found by humans since they're invisible to the naked eye. If one gets clicked, it's most certainly a bot that did it.

Conclusion

Web scrapers are a terrific tool for businesses. They allow business owners to quickly collect highly relevant data that would otherwise cost them time, money, and effort to obtain.

If you or your team don't want to deal with the headache of scraping the web or can't keep up with the internet's anti-bot mechanisms, you can easily purchase a good web scraping bot and perform your data gathering activities yourself.

Our Scraping Robot API is an affordable solution that's easy to use even if you have no programming experience. It can help you extract the data you need with one simple command, and it follows web scraping best practices. Try a demo of any of our modules today. No signup or login information needed!

Submit

Related Articles

How To Pull Data From a Website

Building A Competitive Pricing Strategy With Web Scraping

Web Scraping Food Delivery Data (Why Does It Matter)

Using A Web Scraper API To Revolutionize Data Optimization

Alli Davis

Alli's web scraping insights are excellent resources for beginners and experts alike. With a background in journalism and technical writing, she knows how to ask questions that help people reach their "lightbulb" moments. When it comes to web scraping tips, tricks and ideas, she's got you covered. If you have a blog topic or other scraping idea for Alli, send her a note via the Scraping Robot contact page. Until then, happy scraping!

Help