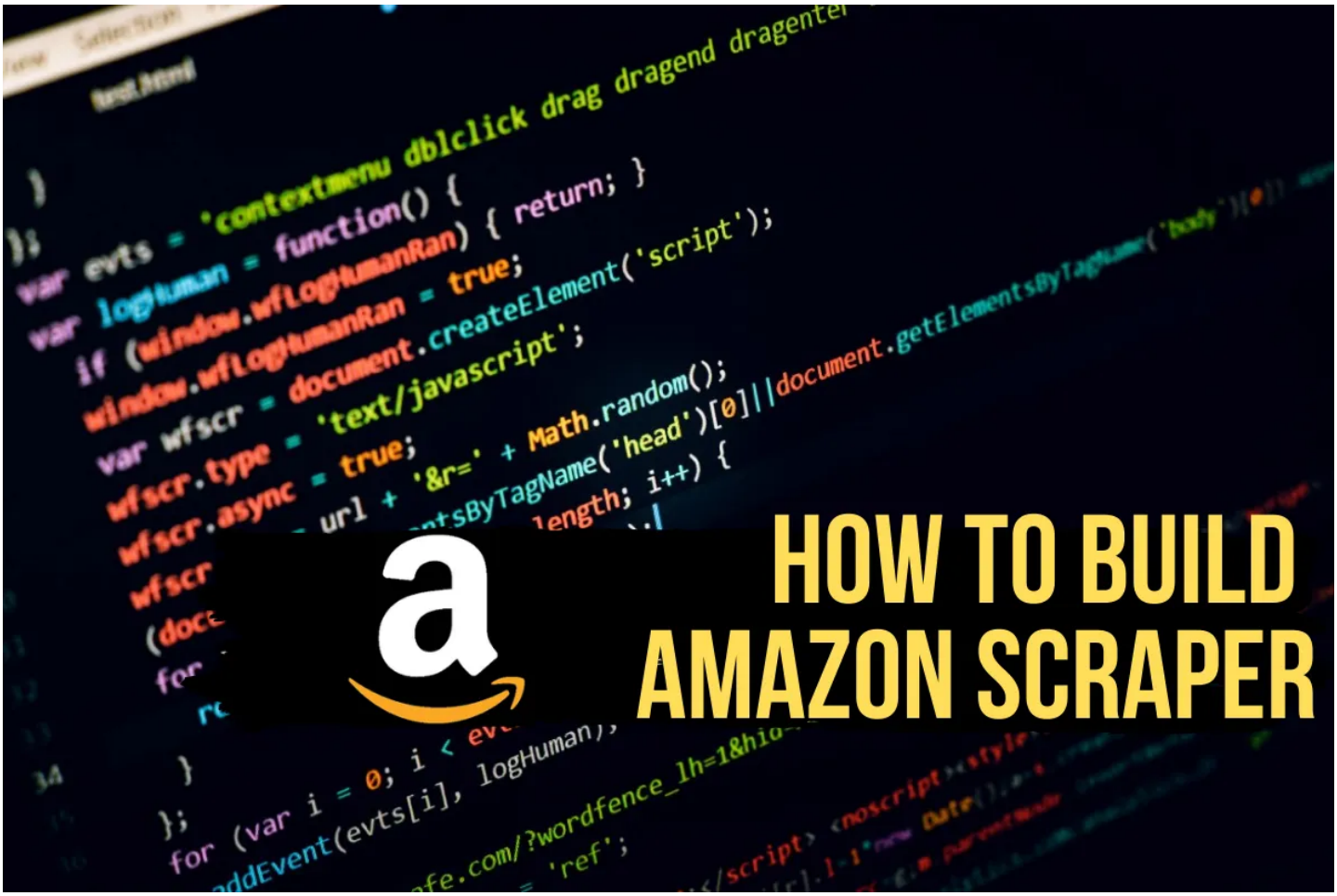
How To Build An Amazon Product Scraper With Bypass Bot Detection

Jevri Malku · Follow

5 min read · Dec 13, 2021



Today I will show you how to build a web scraper for Amazon. This tool will be designed to collect basic information about products from a specific keyword.

> **This is also available on YouTube now. If you learn better with videos, then check it out below.**

Web scraping allows you to extract specific data from the Amazon website and save it in a spreadsheet or JSON file. You could even turn this into an automatic process that updates your data on a daily, weekly, or monthly basis.

· Look for the data we need

So here, we will extract product data information from amazon.com by using 'headphones' as a sample keyword. Then extract the available

information from the search results page, and retrieve the details of each product. The URL for this search and the page we'll be scraping is https://www.amazon.com/s?k=headphones.

data we need

· Required Tools

Let's ensure we have all the following tools installed and configured before continuing to the next step.

· PyCharm -Download from here

· Python 3.9-Download from here

· Getting Started

Create a new project on PyCharm, follow this instruction.

For the code to work, install all the necessary libraries by running pip install on the terminal.

> *pip install requests beautifulsoup4 pandas fake-useragent-ex*

· Explore before Coding

Before we start coding, we have to understand the page's content and structure. For that, the easier way we know is to inspect the target page using a browser. We will be using Chrome's DevTools, but other browsers have similar tools.

result of inspect element

· Scraping Amazon Results Page

After we installed all the dependencies presented above, let's create a new rsult_page.py. and type the following lines of code:

Run the code above, and you will get some product results from page 1.

Now, you might want to scrape several pages worth of data for this project. By adding pagination, add this code to the previous code.

```
last_page = soup.find('li', {'class': 'a-disabled a-last'})
if not last_page:
    pass
else:
    break
```

This is used to check until the last page is found on the search page.

With a few changes, the above code will be combined into a function.

And see when you run this, we have got the results of all the pages.

Output will be like this :

- **Scraping Amazon Product Page**

After getting the results from the search page, then get detailed information from each product.

Type the code below:

Run the code, and see the output :

we will retrieve detailed information such as:

> *Title Product, Price, Rating, Stock, features.*

In short, how to get? see the inspect element of any information we want. I have summarized it will explain a bit, for more details see the whole code at the end.

- **This variable is to get Title Product**

```
title = soup.find('span', attrs={'id': 'productTitle'}).text.strip()
```

- Get the price

```
current_price = soup.find('span', attrs={'class': 'a-price a-text-
price a-size-medium apexPriceToPay'}).find('span', attrs={'class':
'a-offscreen'}).text.strip()
```

- Get review count

```
review_count = soup.find('span', attrs={'id':
'acrCustomerReviewText'}).text.strip()
```

- Get Stock

```
available_stock = soup.find('div', attrs={'id':
'availability'}).find('span').text.strip()
```

- Features

```
feature_bullet = soup.find('div', attrs={'id': 'feature-
bullets'}).find('ul', attrs={'class': 'a-unordered-list a-vertical
a-spacing-mini'}).find_all('li')
```

And our scaper is almost done, from the search results it has been successful, also taking the details of each product. now it's time to combine the above functions into one complete script.

After running the above codes sometimes amazon will block your IP and detect it as a robot — because there are too many requests in a short time. but don't worry, I'll give you the tips.

Add the following function code.

Yes, right by using a random user agent, you can bypass detection as a robot. The process takes a random user agent, then checks it, if it is detected as a robot it will repeat looking for a new user agent until it can be passed.

The complete code above, I made one below and also please check on my github.

Run main.py

======= Update ========

Hi all, maybe when you read this story now, and try it you will get some error. Like “fake_useragent.errors.FakeUserAgentError: No browsers version found for Chrome”

This is due to a problem with the package. I’ve been updated the code. and its work like a charm.

update code scraper.py

==========

The result should look something like this:

> *The post is for learning purposes only in order to demonstrate different techniques of developing web scrapers. The Author does not take responsibility for how the code snippets are used.*

If you find this article helpful share it with your friends and followers and check out my other posts.

Amazon Scraper | Python | Beautifulsoup | Requests Library

Written by Jevri Malku

Follow

10 Followers

Excited about automating daily routines, data scraping, and browser automation.

Recommended from Medium

Builescu Daniel in Python in Plain English

Web Scraping and Web Automation with Python 2023

Introduction to Web Scraping and Web Automation with Python 2023

· 10 min read · Apr 14

111 1

André Ribeiro in Geek Culture

Going Incognito: Scrapy with Proxies and Headers Rotation

Anonymizing Web Scraping with Python and Scrapy

· 7 min read · Feb 21

2

Lists

Coding & Development
11 stories · 85 saves

Predictive Modeling w/ Python
18 stories · 204 saves

Practical Guides to Machine Learning
10 stories · 219 saves

New_Reading_List
174 stories · 51 saves

Kristen Walters in Adventures In AI

5 Ways I'm Using AI to Make Money in 2023

These doubled my income last year

✨ · 9 min read · Jul 19

18.6K 295

Jacob Bennett in Level Up Coding

Use Git like a senior engineer

Git is a powerful tool that feels great to use when you know how to use it.

✨ · 4 min read · Nov 14, 2022

8.5K 85

Tarik Kaoutar (高達烈) in DataDrivenInvestor

How To Use ChatGPT To Automate Web Scraping

🤦🏾‍♂️ GitHub ⭐| 🧿 Twitter | 📹 Linkedin | 🕸 Website | 😃UpWork

✨ · 4 min read · Feb 9

440 7

Dominik Polzer in Towards Data Science

All You Need to Know to Build Your First LLM App

A step-by-step tutorial to document loaders, embeddings, vector stores and prompt...

✨ · 26 min read · Jun 21

3.7K 35

See more recommendations