

# How to Bypass Cloudflare in Python

Updated: July 12, 2024 · 10 min read

## Table of contents

Does your Python scraper get blocked by Cloudflare? That's because you keep getting in the line of the anti-bot. But no worries. You can bypass it and scrape your desired data.

We'll show you 5 tested and trusted Python tools to help you scrape without getting blocked by Cloudflare.

- [Cloudscraper](#).
- [ZenRows](#).
- [Undetected-chromedriver](#).
- [Curl\\_cffi](#).
- [Cfscrape](#).

Let's go!

## Can Cloudflare Detect Python Scrapers?

**Cloudflare is one of the most common anti-bot measures you'll encounter while scraping. It uses various detection techniques with different protection levels to block scrapers.** While some protection levels are simple and easier to break with a few customizations, others are advanced, and bypassing them requires combining several solutions.

Like any other programming language, Python scrapers are automated scripts, so they're prone to Cloudflare's detection.

For example, Python's Requests library, a popular HTTP client, will likely not pass Cloudflare's defense because it sends bot-like parameters like the `python-requests/2.32.3` User Agent. Additionally, it doesn't support JavaScript rendering and lacks the browser-like features to automate human interaction with a website.

To fact-check that claim, let's see how the Requests library performs against a Cloudflare-protected website, [G2 Reviews](#). Try it out with the code below:

### Example

```
# import the required library
import requests

# send the request
response = requests.get("https://www.g2.com/products/jira/reviews")

# validate and print the response
if response.status_code != 200:
    print(f"The request failed with an error {response.status_code}")
else:
    print(response.text)
```

Output



The request failed with an error 403

**Cloudflare keeps updating its defense system**, making it more difficult to bypass. To stay ahead of Cloudflare's detection mechanisms, you'll need to implement specific bypass techniques in your web scraper.

For a detailed guide on different techniques useful for tricking Cloudflare, read our article on [bypassing Cloudflare](#).

Now, let's show you the five solutions to bypass Cloudflare and scrape without getting blocked in Python.

#### Frustrated that your web scrapers are blocked once and again?

ZenRows API handles rotating proxies and headless browsers for you.

[Try for FREE](#)

. . .

## How to Bypass Cloudflare in Python

Different libraries and tools may help bypass Cloudflare while [web scraping in Python](#). Let's look at the five best examples and learn how each works.

### Cloudscraper

[Cloudscraper](#) was built as an easy-to-use browser emulator for bypassing Cloudflare in Python. It's similar to the Requests library in functionality and parameter acceptance. Cloudscraper's JavaScript engine makes it possible to easily decode and parse JavaScript, allowing your request to imitate a regular web browser's behavior.

Cloudscraper supports different browsers, including Chrome and Firefox, and emulates fingerprints, such as the cipher suites, for a secure client-server connection.

However, the downside of using Cloudscraper is that it doesn't pass advanced fingerprinting tests, making it unfit to [evade sophisticated Cloudflare Turnstile CAPTCHA](#). That said, pairing it with paid CAPTCHA solvers like [2Captcha](#) and fortifying it with proxies can enhance its evasion capability.

#### 👍 Pros

- Easy to use.
- Features [User Agent auto-rotation for Cloudflare bypass](#).
- Suitable for bypassing basic blocks.

#### 👎 Cons

- It fails on websites using advanced Cloudflare protection.

Check out our detailed tutorial on [using Cloudscraper with Python](#) to learn more.

## ZenRows

The best way to bypass Cloudflare with Python is to use [ZenRows](#). It's a **web scraping solution that bypasses Cloudflare with a single API request**. As a full-fledged bypass toolkit, ZenRows allows you to focus on your scraping logic while it handles anti-bot auto-bypass under the hood.

In addition to a scraping API, ZenRows also features residential proxy services with geo-targeting and auto-rotation features. If scraping a JavaScript-rendered website, it acts as a headless browser to automate human actions such as scrolling, clicking, and more. ZenRows also works with any programming language and is compatible with other libraries, making it easy to integrate into your existing workflows.

### 👍 Pros:

- Easy to use.
- Compatible with any programming language
- Bypasses Cloudflare and other anti-bot measures, regardless of the difficulty level.
- It features smart rotating premium proxies.
- Scrapes JavaScript-rendered pages.
- Easy to integrate into your existing workflows.
- 24/7 customer support with a complete knowledge base.
- Frequently updated.

### 👎 Cons:

- It's a paid service (but offers a free trial).

## How to Bypass Cloudflare in Python Using ZenRows

Let's use ZenRows to scrape [G2 Reviews](#), a website heavily protected by Cloudflare, to see how it works. To bypass Cloudflare with ZenRows, you only need your free API key and the target URL.

[Sign up](#) to open the ZenRows Request Builder. Paste the target URL in the link box, activate Premium Proxies, and click JS Rendering. Choose Python as your programming language and select the API connection mode. Now, copy and paste the generated code into your scraper file.

The screenshot displays the ZenRows Builder interface. On the left, a sidebar contains navigation links for PRODUCTS (Scraper API, Residential Proxies, Scraping Browser), ANALYTICS (Overview), BILLING (Plans), and Docs (Settings, 0% usage). The main area is titled 'Builder' and features a 'Build Your Request' panel. This panel includes a 'URL to Scrape' field with the value 'https://httpbin.io/anything', a 'Boost mode' section with 'AI Web Unblocker' and 'Premium Proxies' (checked), and 'Advanced settings' with 'JS Rendering' (checked). Below these is a 'Try it' button and a note 'This request costs \$0.001'. To the right, the 'Integrate ZenRows Into Your Codebase' section shows an 'API Key' field and a list of programming languages (Python, Node.js, Java, PHP, Go, Ruby, C#, cURL). The Python option is selected, and a code block displays the generated Python code for making a request to the ZenRows API.

```
# pip install requests
import requests

url = 'https://httpbin.io/anything'
apikey = <your_zenrows_api_key>
params = {
    'url': url,
    'apikey': apikey,
}
response = requests.get('https://api.zenrows.com/v1', params=params)
print(response.text)
```

Click to open the image in full screen

## Example

```
# pip install requests
import requests

url = 'https://www.g2.com/products/jira/reviews'
apikey = '<YOUR_ZENROWS_API_KEY>'
params = {
    'url': url,
    'apikey': apikey,
    'js_render': 'true',
    'premium_proxy': 'true',
}
response = requests.get('https://api.zenrows.com/v1/', params=params)
print(response.text)
```

The above code accesses the protected web page and extracts its full-page HTML, as shown:

## Output

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <link href="https://www.g2.com/assets/favicon-fdac..." rel="shortcut icon" type="image/x-
icon" />
  <title>Jira Reviews from July 2024</title>
</head>
<body>
  <!-- other content omitted for brevity -->
</body>
```

Great! Your scraper now uses ZenRows to bypass Cloudflare.

## Undetected-chromedriver

Undetected-chromedriver is a modified version of the **Selenium** ChromeDriver that prevents anti-bot detection. The module automatically loads a ChromeDriver binary into your machine and patches it to emulate a legitimate browser's fingerprint. If you want to learn about bypassing Cloudflare using Selenium, don't forget to check out our guide.

One of the main advantages of Undetected ChromeDriver is that it supports JavaScript execution to interact dynamically with websites. This feature adds a human touch to your request and boosts your chances of evading Cloudflare.

However, like other headless browsers, undetected Chromedriver introduces memory overhead since it triggers a browser instance. Although it may get blocked with more sophisticated Cloudflare protection, you can add proxies to undetected-chromedriver to give it more stealth.

### 👍 Pros:

- Increases the chances of bypassing browser fingerprinting tests.
- Ability to mimic human behavior.
- Suitable for handling JavaScript challenges.

### 👎 Cons:

Read our complete tutorial on [how to use the undetected-chromedriver](#) in Python to learn more.

## Curl\_cffi

[Curl\\_cffi](#) is an improved version of the standard cURL library in Python. It patches the standard cURL library with actual browser fingerprints, allowing it to emulate popular browsers like Chrome, Safari, and Edge.

It bypasses anti-bots like Cloudflare by replacing detectable bot-like signals, such as cURL's OpenSSL library, with Chrome's BoringSSL. However, curl\_cffi isn't a headless browser and can't execute JavaScript. So, it has fingerprinting limitations and can still be flagged by advanced anti-bot measures.

### 👍 Pros:

- Emulates different browsers without introducing memory overhead.
- Good for bypassing basic fingerprinting tests.
- Simple learning curve.

### 👎 Cons:

- It can't bypass sophisticated anti-bot measures.
- Its inability to execute JavaScript reduces the chances of evading anti-bots.

Want to learn more about curl\_cffi? **Check out our detailed tutorial on [using curl\\_cffi for Python web scraping](#).**

## Cfscrape

The [cfscrape](#) library is another popular browser emulator for bypassing Cloudflare in Python. This tool follows the same operational pattern as Cloudscraper. The only difference is that cfscrape doesn't allow you to specify a browser, as it only emulates Chrome by default.

That said, cfscrape isn't perfect: **it can only handle webpages with simple anti-bot measures**, meaning it's ineffective against advanced Cloudflare protection. It also requires downgrading to a lower urllib3 version because it doesn't support urllib3 versions 2+. You might need to run the following command to install a specific lower version of urllib3:

Terminal



```
pip install "urllib3<2"
```

### 👍 Pros:

- Easy to use and implement.

### 👎 Cons:

- Ineffective against advanced Cloudflare challenges.
- It lacks maintenance and updates.
- It's less feature-rich than other options like Cloudscraper.
- Incompatible with recent urllib3 versions (versions 2+).

on [using ciscrape with Python](#).

. . .

## Conclusion

Knowing how to bypass anti-bots is as vital as the scraping process itself, especially when you're looking to scrape a web page protected by Cloudflare. In this article, we covered 5 techniques for bypassing Cloudflare using Python. They include Cloudscraper, ciscrape, ZenRows,undetected-chromedriver, and curl\_cffi.

While the other tools often fail with large-scale scraping or advanced Cloudflare security measures, ZenRows is the only solution capable of bypassing Cloudflare and any other anti-bot at scale. What's more, ZenRows only requires a single API call and doesn't involve technical setup like the other tools.

[Try ZenRows for free now](#) without a credit card!

## Frequent Questions

What Is Cloudflare Bot Manager?



Share



## Ready to get started?

Up to 1,000 URLs for free are waiting for you

[Try ZenRows for Free](#)



ZenRows® web scraping API handles all anti-bot bypass for you with rotating proxies, headless browsers and more.



[Pricing](#)[API Documentation](#)[Legal](#)[Affiliate](#)[Contact](#)[Press](#)[Customers](#)[Careers](#)

# Scraping Tutorials


[Web Scraping in Python](#)[Web Scraping in NodeJS](#)[Web Scraping in Java](#)[Web Scraping in PHP](#)[Web Scraping in R](#)[Web Scraping in Ruby](#)[Web Scraping in Golang](#)[Web Scraping in C#](#)[Web Scraping in Rust](#)[Web Scraping in C++](#)[Web Scraping in C](#)[Perl Web Scraping](#)[Scrapy Python Web Scraping](#)[cURL Converter](#)[Selenium Web Scraping](#)[Playwright Web Scraping](#)[Puppeteer Web Scraping](#)[Zillow Scraper](#)[Indeed Scraper](#)[All Popular Web Scrapers](#)

# Anti-bot

[Bypass Cloudflare](#)[Bypass Akamai](#)[Bypass PerimeterX](#)[Bypass DataDome](#)[Web Scraping Without Getting Blocked](#)[Avoid Getting Blocked in Python](#)[Solve CAPTCHAs](#)[Web Scraping Proxy](#)

# How we compare

[ScrapingBee Alternative](#)[BrightData Alternative](#)[ZenRows Alternative](#)

 All services are online

2025 ZenRows®. All rights reserved.

[X](#) [in](#)