

Forum:[Hack Forums](#)

Thread: [what is the easiest way to web automation](#)

flyingduck - August 23, 2017, 10:28 am

hey guys,

i want to learn basic web automation, i have no prior experience and i need a copy&paste solution. something like a macro but maybe a bit more intelligent. what can you recommend me? thanks!

Anti - September 13, 2017, 12:11 am

Use a language like Python for this.

shhhark - November 14, 2017, 11:37 am

+1 for Python

If you need to tackle JavaScript web elements - use Selenium!
otherwise - requests + beautifulsoup

this is my OPINION, there are dozen ways to do the same thing sooo

Effexion - November 14, 2017, 11:40 am

You could do this with a VB webbrowser and a small amount knowledge. Why C&P? Take this opportunity to learn, man. I'm a noob myself but it's pretty fun to learn and massively rewarding when you do your task from your own knowledge.

Hapi - November 15, 2017, 8:25 am

Do you have any experience at all with programming and the web? There are quite a lot of options you can go for, but you may want to learn the basics of programming first before you jump right into it.

Scrapy (<https://scrapy.org/>)

"An open source and collaborative framework for extracting the data you need from websites. In a fast, simple, yet extensible way."

Python based. Supports 2.* and 3.*.

Runs on Windows, Linux and MacOS.

Works with HTTP proxies.

Basic HTTP Authentication.

Example code: Here's the code for a spider that scrapes famous quotes from website <http://quotes.toscrape.com>, following the pagination:

Code import scrapy

```

class QuotesSpider(scrapy.Spider):
    name = "quotes"
    start_urls = [
        'http://quotes.toscrape.com/tag/humor/',
    ]

    def parse(self, response):
        for quote in response.css('div.quote'):
            yield {
                'text': quote.css('span.text::text').extract_first(),
                'author': quote.xpath('span/small/text()').extract_first(),
            }

        next_page = response.css('li.next a::attr("href")').extract_first()
        if next_page is not None:
            yield response.follow(next_page, self.parse)

```

Selenium (<http://www.seleniumhq.org/>)

"Selenium automates browsers. That's it! What you do with that power is entirely up to you. Primarily, it is for automating web applications for testing purposes, but is certainly not limited to just that"

Works with the following languages: Java, C#, Python, Ruby, PHP, Perl and JavaScript.

Create tests for any websites.

Automate purchases on stores.

This is a basic Selenium java example. It uses the default HtmlUnitDriver to extract the page title in console-like style:

```

Code import org.openqa.selenium.WebDriver;
import org.openqa.selenium.htmlunit.HtmlUnitDriver;
public class SeleniumConsoleExample {
    public static void main(String[] args) {
        // Create HTML Unit Driver - this is the build in Selenium client
        WebDriver driver = new HtmlUnitDriver();
        // go to url
        driver.get("http://javatutorial.net");
        // Check the title of the page
        System.out.println("Page title is: " + driver.getTitle());
        driver.quit();
    }
}

```

If you are wanting to create bots (sneaker bots, etc) then you may want to look into creating your own bot with another language, such as; Go, C/C++, Java, etc. Making it a separate program means that you are going to have to handle logging in to websites, understanding how those websites work and making your bot work with that, and when the website changes you are going to need to fix that.

I'm pretty sure most custom bots are written in C++, generally for performance - if 1,000 people are trying to buy from a website then you are going to need a fast bot. I'm not sure how tools like Selenium perform, but if you are just looking to play a game or create bots for other websites then of course use the tools already available.