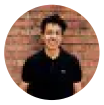
Data

Web Scraping using Puppeteer and Node

June 10, 2019



Hrithik Jha
Software Engineer

Introduction

Puppeteer is a node library which provides an API to control Google Chrome and Chromium. It can be used to scrape all aspects of a Chrome (or Chromium) window including the Chrome Developer Tools. Today, we'll be scraping lobste.rs.

Why Scrape Data?

Scraping can be used for many different purposes. It helps collect data for Machine Learning or can be used for Data Visualization. It can also be used to automate processes. An amazing example would be to scrape movie tickets and prices of shows in theaters nearby. With the data you can sort in order of the price, show timing even with data from different websites to make decisions.

Getting Started

For starters, you'd need a text editor (Pro Tip: MS Word with Consolas) and a system with Node. You can get Node from here or you can use `brew install node` on a Mac, or `sudo apt-get install nodejs npm` on Linux.

To install Puppeteer, simply run:

```
npm install puppeteer
```

Puppeteer is built on Node and can be run with any existing node program. One might use Requests or Selenium (more scraping and automation libraries) with Flask or Django, similarly we can use Puppeteer for programs written on Node.

Into The Code

The first thing we do is to import the Puppeteer library to be used in the program.

```
const puppeteer = require('puppeteer');
```

The scraping logic is written inside the appropriately named function, `scrape()`. You would also notice the async keyword. That is needed as most of the Puppeteer API are asynchronous and have to be written inside an async function.

```
let scrape = async () => {
```

Inside the function, we initialize the browser and the page. It is very similar to the flow of opening a browser and website as one would, open browser > open new tab > go to the URL.

```
const browser = await puppeteer.launch({headless: false});
const page = await browser.newPage();
await page.goto('https://lobste.rs/');
```

We have a delay to allow the page load the DOM contents as our queries would not work if the element we want to scrape has not loaded.

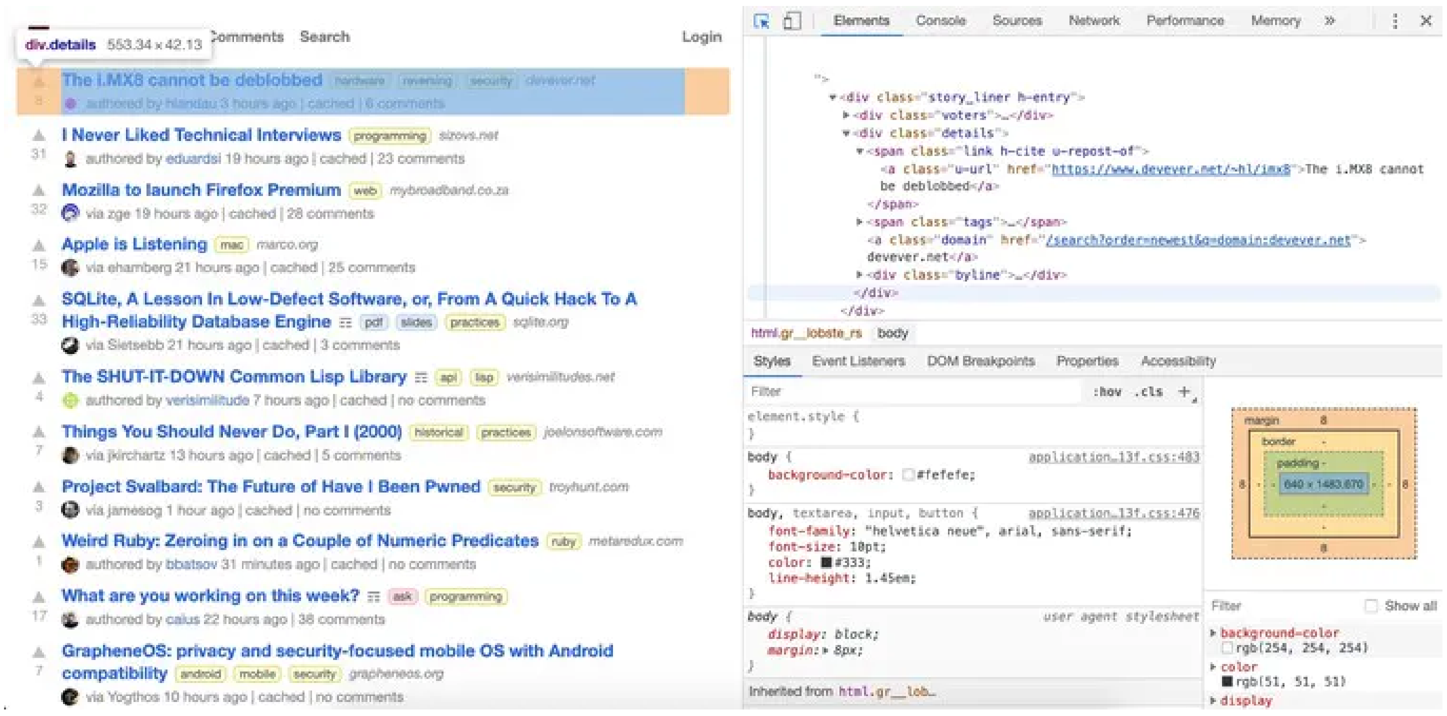
```
await page.waitFor(1000);
```

In Puppeteer, all interaction with the website's content is done using the page.evaluate function.

```
const result = await page.evaluate(() => {
```

Inside the function, we initialise an empty array to store the scraped data.

Here, we save the elements with the class name `u-url` and `score`. To get the class name of the element you want to scrape press Ctrl + Shift + I on Windows or Cmd + Opt + I on Mac. Here, click the top left button to select an element. Once that element is highlighted in the Elements tab, right click on it and copy the class name of the element.



div.details 553.34 × 42.13

Comments Search Login

8 The i.MX8 cannot be deblobbed hardware reversing security devever.net
authored by hlandau 3 hours ago | cached | 6 comments

31 I Never Liked Technical Interviews programming sizovs.net
authored by eduardsi 19 hours ago | cached | 23 comments

32 Mozilla to launch Firefox Premium web mybroadband.co.za
via zge 19 hours ago | cached | 28 comments

15 Apple is Listening mac marco.org
via ehamberg 21 hours ago | cached | 25 comments

33 SQLite, A Lesson In Low-Defect Software, or, From A Quick Hack To A High-Reliability Database Engine pdf slides practices sqlite.org
via Sietsebb 21 hours ago | cached | 3 comments

4 The SHUT-IT-DOWN Common Lisp Library api lisp verisimilitudes.net
authored by verisimilitude 7 hours ago | cached | no comments

7 Things You Should Never Do, Part I (2000) historical practices joelonsoftware.com
via jkirchartz 13 hours ago | cached | 5 comments

3 Project Svalbard: The Future of Have I Been Pwned security troyhunt.com
via jamesog 1 hour ago | cached | no comments

1 Weird Ruby: Zeroing in on a Couple of Numeric Predicates ruby metaredux.com
authored by bbatsov 31 minutes ago | cached | no comments

17 What are you working on this week? ask programming
authored by calus 22 hours ago | 38 comments

7 GrapheneOS: privacy and security-focused mobile OS with Android compatibility android mobile security grapheneos.org
via Yogthos 10 hours ago | cached | no comments

Elements Console Sources Network Performance Memory »

```
">
▼<div class="story_liner h-entry">
  ▶<div class="voters">…</div>
  ▼<div class="details">
    ▼<span class="link h-cite u-repost-of">
        <a class="u-url" href="https://www.devever.net/~hl/imx8">The i.MX8 cannot
        be deblobbed</a>
      </span>
    ▶<span class="tags">…</span>
      <a class="domain" href="/search?order=newest&q=domain:devever.net">
      devever.net</a>
    ▶<div class="byline">…</div>
    </div>
  </div>
```

html.gr__lobste_rs body

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

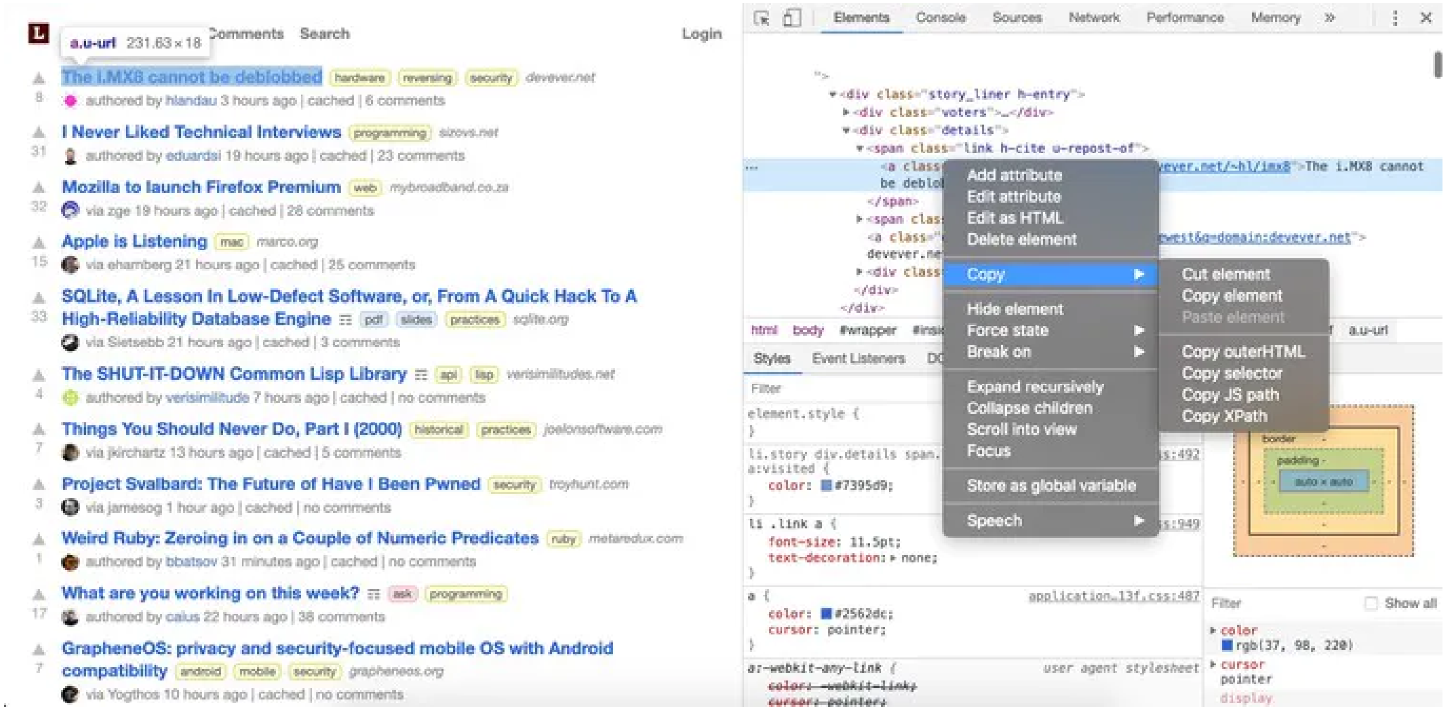
```
element.style {
}
body {                                   application…13f.css:483
  background-color: #fefefe;
}
body, textarea, input, button {          application…13f.css:476
  font-family: "helvetica neue", arial, sans-serif;
  font-size: 10pt;
  color: #333;
  line-height: 1.45em;
}
body {                                   user agent stylesheet
  display: block;
  margin: ▸ 8px;
}
Inherited from html.gr__lob…
```

margin 8 border - padding - 640 × 1483.670

Filter Show all

```
▸ background-color
    rgb(254, 254, 254)
▸ color
    rgb(51, 51, 51)
▸ display
```

In this example we use Class Names, but there are many other methods such as using the CSS Selector, the element ID or the XPath. The JavaScript Document object has support for using these methods as well.



a.u-url 231.63 × 18

Comments Search Login

8 The i.MX8 cannot be deblobbed hardware reversing security devever.net
authored by hlandau 3 hours ago | cached | 6 comments

31 I Never Liked Technical Interviews programming sizovs.net
authored by eduardsi 19 hours ago | cached | 23 comments

32 Mozilla to launch Firefox Premium web mybroadband.co.za
via zge 19 hours ago | cached | 28 comments

15 Apple is Listening mac marco.org
via ehamberg 21 hours ago | cached | 25 comments

33 SQLite, A Lesson In Low-Defect Software, or, From A Quick Hack To A High-Reliability Database Engine pdf slides practices sqlite.org
via Sietsebb 21 hours ago | cached | 3 comments

4 The SHUT-IT-DOWN Common Lisp Library api lisp verisimilitudes.net
authored by verisimilitude 7 hours ago | cached | no comments

7 Things You Should Never Do, Part I (2000) historical practices joelonsoftware.com
via jkirchartz 13 hours ago | cached | 5 comments

3 Project Svalbard: The Future of Have I Been Pwned security troyhunt.com
via jamesog 1 hour ago | cached | no comments

1 Weird Ruby: Zeroing in on a Couple of Numeric Predicates ruby metaredux.com
authored by bbatsov 31 minutes ago | cached | no comments

17 What are you working on this week? ask programming
authored by calus 22 hours ago | 38 comments

7 GrapheneOS: privacy and security-focused mobile OS with Android compatibility android mobile security grapheneos.org
via Yogthos 10 hours ago | cached | no comments

Elements Console Sources Network Performance Memory »

```
">
▼<div class="story_liner h-entry">
  ▶<div class="voters">…</div>
  ▼<div class="details">
    ▼<span class="link h-cite u-repost-of">
        <a class=  …  ever.net/~hl/imx8">The i.MX8 cannot
        be deblo
      </span>
    ▶<span clas
      <a class="    …  west&q=domain:devever.net">
      devever.ne
    ▶<div class
    </div>
  </div>
```

html body #wrapper #inside … a.u-url

Styles Event Listeners

```
element.style {
}
li.story div.details span.              …s:492
a:visited {
  color: #7395d9;
}
li .link a {                            …s:949
  font-size: 11.5pt;
  text-decoration: ▸ none;
}
a {                                      application…13f.css:487
  color: #2562dc;
  cursor: pointer;
}
a:-webkit-any-link {                     user agent stylesheet
  color: -webkit-link;
  cursor: pointer;
```

border - padding - auto × auto

Filter Show all

```
▸ color
    rgb(37, 98, 220)
▸ cursor
    pointer
  display
```

```
let articles = []
let titles = document.getElementsByClassName('u-url');
let scores = document.getElementsByClassName('score');
```

After copying it, you can then change the parameter in `.getElementsByClassName` to get the data in that element.

Next, we write a for loop to save all Titles and Scores in the Articles array. Before doing that we check if the number of Titles and Scores we have are the same to avoid any logical errors.

```
if(titles.length == scores.length) {
            for (i = 0; i < titles.length; i++) {
                articles.push(
                {
                    'title': titles[i].textContent,
                    'score': scores[i].textContent
                })
            }
        }

        return {
            articles
        }

    });
```

After scraping, we close the browser and return the Articles.

```
    browser.close();
    return result;
};
```

We return Articles, to the caller of the function. Here, we have used `.then()` to print the values in console only after the function is done running.

Clubhouse

```
    console.log(value);
});
```

Conclusion

We finally have all the scraped data in the console. This can further be modified to save the data in an Excel sheet or a Json file. Apart from scraping Puppeteer can be used to automate processes like submitting assignments or filling forms online.

Thank you for reading the article!

Share this Developer How-to

3 Comments **Clubhouse** 1 **Login**

Recommend Tweet Share **Sort by Best**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Brent Engelbrecht • 7 months ago

Thanks! Awesome, quick 'n easy tutorial.

• Reply • Share ›

James Vreeken • 8 months ago • edited

You could also use webtask.io if you can't install node on your computer for whatever reason... or if you wanted to set it on a cron job to run every so often.

Edit: doesn't seem to work as webtask is serverless and doesn't have chrome installed.

I've used webtask.io with parsehub's webhooks to automate and its worked pretty well so I was interested to try this out.

Question:
Instead of waiting an arbitrary 1000ms the "page.goto" method is supposed to wait for the DomContentLoaded event: https://github.com/GoogleCh...

There's also networkidle0 ? any thoughts on this so we don't have to arbitrarily pick a random number of time we expect it to take to load?

• Reply • Share ›

Hrithik Jha ➦ James Vreeken • 7 months ago

Hi James, I greatly appreciate your response. I mostly use Puppeteer for automation and web scraping is a

Join the over 25,000 developers, designers, and product managers already using Clubhouse

Get started—free forever

Sign up with Google

Product

Features
Pricing
Enterprise
Integrations
Write beta

Company

About Us
Customers
Careers
Press
Contact
Brand Guidelines
FAQ

Developers

REST API Docs
Webhook API Docs
Community
Open Source Projects
Developer How-Tos

Resources

Help Center
Blog
Webinars
Status
Referral Program