

Open in app ↗

Sign
up

Sign In



Photo by [Rock'n Roll Monkey](#) on [Unsplash](#)

How to make a Medium reader bot with Puppeteer

And much more !!!



Nowshad

Follow

Published in
codeburst

7 min read

Jun 6, 2019



Listen



Share

So, on an awesome Thursday afternoon, I was thinking for a quick side project in my office (which we are encouraged to do). I'm an avid reader of Medium.com, and I write sometimes as well. So, I was messing with Medium's view count and wanted to understand how that actually works!

So, after a few manual trial and errors, I got a few assumptions —

- An article must be opened
- Scrolled to the bottom of the article
- Needs a minimum time to scroll down

So, I started writing a bot with **Google Chrome's headless Puppeteer**. Initially, steps are as simple as the above points. So, let's do that first.

Setup and Installations

I assume you have **NodeJS** installed on your machine. Now, run the following command to generate a fresh NodeJS project -

```
> mkdir medium-reader
> cd medium-reader
> npm init --yes
```

This will create an empty project with only a `package.json` file that's basically a configuration of our

project.

Now, let's install Puppeteer into our project with the command below —

```
npm install puppeteer --save
```

This is the only thing we need to install for now

Hello world with Puppeteer!

Create an `index.js` file at the root of your project directory (i.e. medium-reader) by running `touch index.js`. Now open the project with your favorite code editor (mine is **VS Code**) and paste the sample code below -

Now, we need to run this file with the command `node index.js`.

After this executes, you'll find a file named `example.png` in your projects root directory which will look like the image below —

Screenshot of example.com

What's happening here!

So, in these few lines of code, a lot is happening here —

For easier understanding let's just update the line from

```
const browser = await puppeteer.launch();
```

to

```
const browser = await puppeteer.launch({ headless: false });
```

This will open the puppeteer instance with UI.

puppeteer in action

1. `const puppeteer = require('puppeteer');` imports puppeteer to our project.
2. `const browser = await puppeteer.launch();` This will launch a **chromium instance**. Adding `{ headless: false }` opens chromium with an actual **UI**. Otherwise, it acts as a **command line tool**.
3. `const page = await browser.newPage();` Opens a new **tab** on your browser.
4. `await page.goto('https://example.com', { waitUntil: "networkidle2" });` First, opens the URL `https://example.com` in the current tab, then waits until the network is idle, which means the page has loaded completely. If `{ waitUntil: "networkidle2" }` is not given, the page will not wait for any further loading of assets or other network-related resources.
5. `await page.screenshot({ path: 'example.png' });` Takes a screenshot of the page and saves it as `example.png` in our current directory.

6. `await browser.close();` closes the browser window.

Let's go to Medium.com

As we now understand the steps of using puppeteer, we can now visit our target site — [Medium.com](https://medium.com). we want our articles to be read, we need to find an article or response that's been authored by ourselves to check if the read-view count gets updated or not. So, I target one of my not so famous articles —

Ocean, Mountains, Clouds and Myself : A traveler's evening

A peaceful afternoon at the longest sea beach of the world

medium.com



At the time of writing this tutorial, my above article had **92 views** and **72 Reads** with a view/read ratio of 78%.

Now, I'll target this article for my medium-reader. Let's get into that -

Running this with `node index.js` should open a chromium window and load the article titled — **Ocean, Mountains, Clouds and Myself: A traveler's evening** and close the window when the article is loaded.

Now, we need to scroll to the bottom of the article. Let's write a `autoScroll` function that'll scroll the loaded page with a slow and human speed.

The function might look like a bit too much at first sight but let's break it down. The `page.evaluate` function evaluates the current page and inside it, we are basically scrolling the page for `100px` every `300ms` till we reach the end of the page determined by page height.

Now, let's update our main function and make use of `autoScroll` -

Here, we've changed only two things —

- `page.setViewport` sets the viewport to a standard size.
- Calling `autoScroll` function scrolls the page.

So, our bot is opening and reading my article properly. Now, I can already see my views and reads being incremented as I am testing. So, our basic version of the medium-reader is working !!!

Let's make it SOLID!

Now, what if Medium blocks or doesn't count the read or views anymore when we are sending too many requests from the same source? Our bot will fail in that case. So, let's take a few more steps to make it more fail proof —

- Launch puppeteer in incognito mode.
- Use proxies to bypass any IP detection issues.

Doing both is pretty easy and needs only a little update to our code -

This is straightforward, but **from where I've got that proxy address!!!**

So, if you search for *free web proxy addresses* a lot of proxies will come up, sites like <https://free-proxy-list.net/> enlist a lot of free proxy addresses that are based on different location or anonymous. So, we can pick any one or multiple proxies (prefer elite proxies because they have very good anonymity settings) for our bot. So, instead of setting a fixed proxy we can use a pool of proxy addresses to make sure we are not hitting the URL using the same proxy every time.

Let's add some proxies to our codebase and the content of the file should include code like below —

So, here's a function named `randProxy` which just returns a random proxy address from the above `proxies` array. So, let's use it -

So, we are using the function `randProxy` when launching puppeteer. As a result, every time we launch a new instance of puppeteer, we are basically using a proxy with it.

Now, our bot is more failproof

Enhancements

- When running actual bot and not testing, you **MUST** use `{ headless: true }` param to launch puppeteer instance as you don't wanna see the UI and waste a lot of resources on rendering everything in your browser window.
- You can open multiple instances of puppeteer and run them at the same time.
- Maybe open more than one tab on the same instance which will multiply the page visits like — instance number X tabs number. Though a lot of tabs on the same window won't get counted. So, you need to keep that number low.

- You might need to use an array of articles or links to be read. So, use the same type of implementation as `randProxy` .
- You can add some delay if you want to, between windows or tabs or visits.
- This technique can be used for a lot of use cases as well as other sites for more complex tasks like authentication, crawling, etc.

. . .

Sample source code:

. . .

WARNING ⚠

AS THIS WAS JUST A DEMONSTRATION OF HOW TO MAKE A BOT WITH PUPPETEER AND USE FOR EDUCATIONAL PURPOSE, DON'T USE IT TO SPAM OR GENERATE UNUSUAL TRAFFIC IN MEDIUM.COM OR ANY OTHER SITE. IF YOU DO, AUTHOR WON'T BE RESPONSIBLE FOR THE CONSEQUENCES.

**Thank you for reading,
Have a great day**

JavaScript

Bots

Automation

Programming

Puppeteer



--



2



Follow



Written by Nowshad

546 Followers

.

Dev | Traveler

More from Nowshad and codeburst



Nowshad
in
InfancyIT

KISS => Keep It Simple, Smart

10 Javascript habits to make life easier

4 min read

Feb 17, 2018



--



5



How many String Objects are created here?

```
String s = new String("Java");
```

- 1) One
- 2) Two
- 3) Three



javinpaul
in
codeburst

Top 50 Java Interview Questions for Beginners and Junior Developers

A list of frequently asked Java questions and answers from programming job interviews of Java developers of different experience.

10 min read

.

May 20, 2019

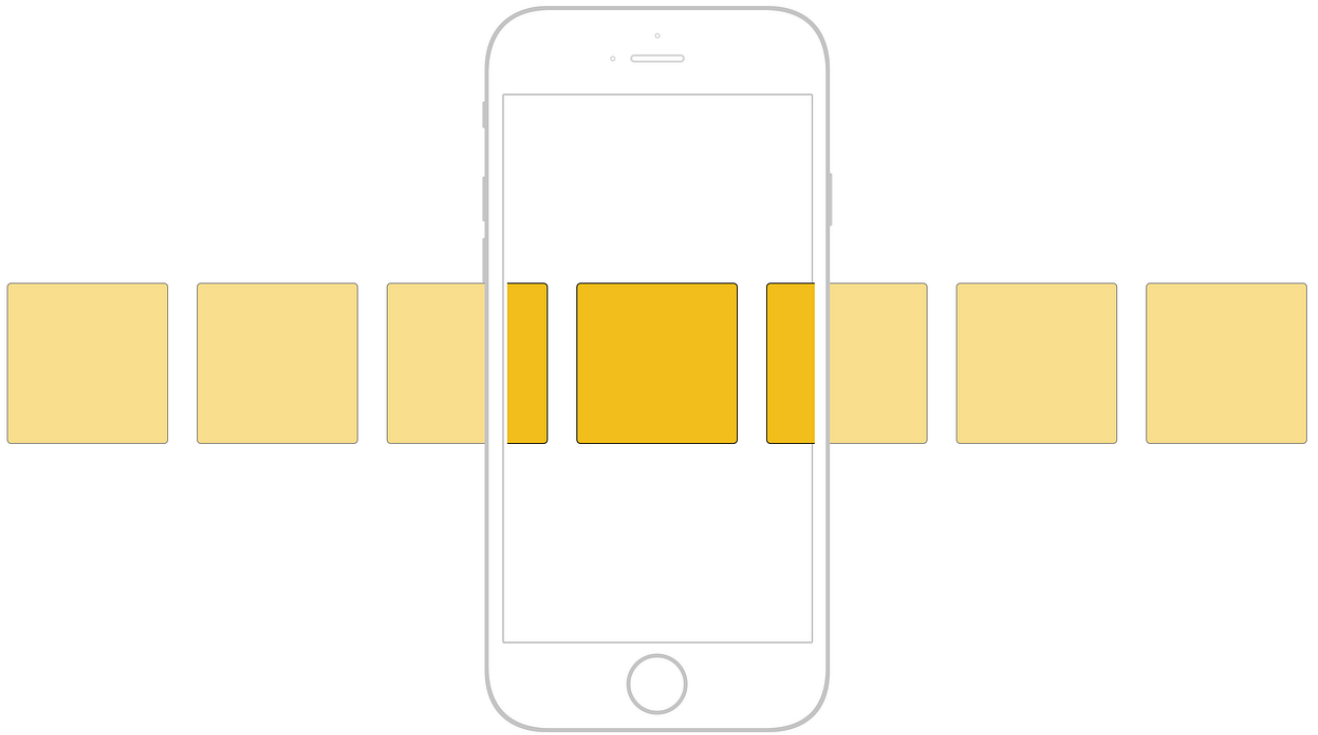


--



7





Colin...

in
codeburst

How To Create Horizontal Scrolling Containers

As a front end developer, more and more frequently I am given designs that include a horizontal scrolling component. This has become...

4 min read

Mar 27, 2017



--



47





Nowshad
in
codeburst

7 Super Useful Aliases to make your development life easier

Let's talk to your machine in your own language

3 min read

Mar 9, 2018



--



6



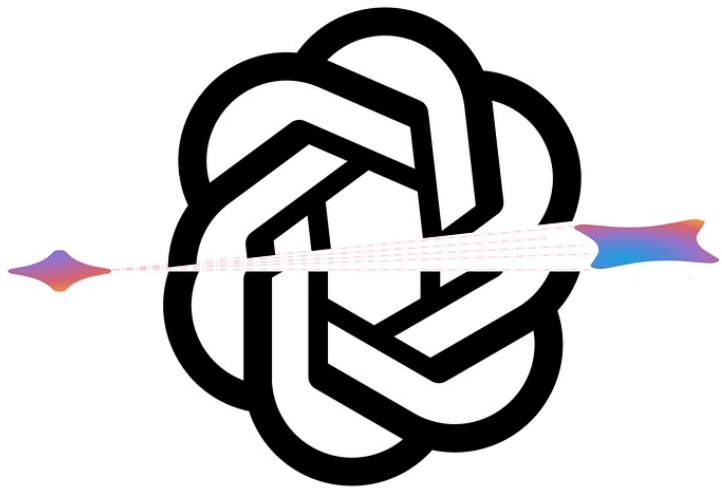
See all from Nowshad

See all from codeburst

Recommended from Medium



Bard



AL Anany



The ChatGPT Hype Is Over—Now Watch How Google Will Kill ChatGPT.

It never happens instantly. The business game is longer than you know.



.

6 min read

.

Sep 1



--



233



10 Apps



Clark

in

Mac O'Clock

Top 10 Must-Have Apple Apps—September 2023

Apps to Elevate Your Health, Enhance Your Mac, AR for Your Car, and More!



13 min read

6 days ago



--



12



Lists





It's never too late or early to start something

15 stories

128 saves



General Coding Knowledge

20 stories

351 saves



Stories to Help You Grow as a Software Developer

19 stories

379 saves



ocker
Science
ects



Coding & Development

11 stories

179 saves



Maximilian Vogel
in
MLearning.ai

The ChatGPT list of lists: A collection of 3000+ prompts, examples, use-cases, tools, APIs...

Updated Sep-09, 2023. Added new prompt engineering courses, masterclasses and tutorials.

10 min read

Feb 7



--



110



Jerameel Delos Reyes...

Moving away from Raspberry Pi as my home server

I use a Raspberry Pi as my home server for a couple of years now. It was a cost-effective and energy-efficient way to host a variety of...

3 min read

Sep 1



--



2



Unbecoming

10 Seconds That Ended My 20 Year Marriage

It's August in Northern Virginia, hot and humid. I still haven't showered from my morning trail run. I'm wearing my stay-at-home mom...



.

4 min read

.

Feb 16, 2022



--



937





Nick Hilton

The End of the Subscription Era is Coming

You're overpaying for your porn (and journalism)

10 min read

Aug 30



--



189



See more recommendations