

Why does phantomJS 2.1.1 render this page as background color with no content, but foxshot 1.2.0 renders it correctly?

Asked 3 years ago Modified 3 years ago Viewed 96 times



1



I like to keep track of Massachusetts lottery results (make sure the results are really random). I can bring up the page of results in any browser and copy-paste the lines of text, and run the text through a filter written in php or gawk to extract the numbers for each date. But I thought it would be nicer to automate the process using phantomJS to get the dates and numbers directly from the DOM.

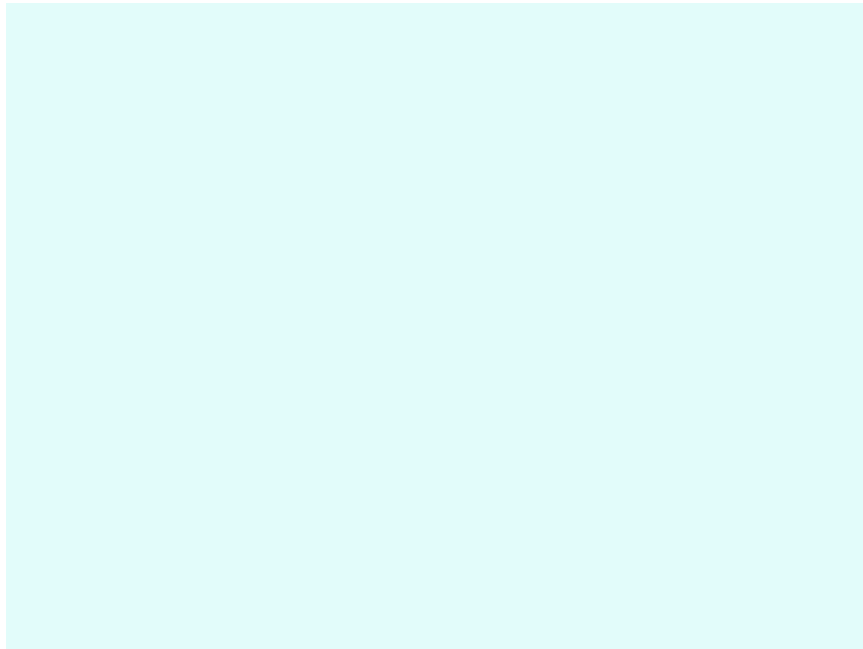


I haven't been able to make this work because the elements containing the results do not appear to be in the DOM accessible through phantomJS. I don't know if I'm doing something wrong (I probably am) or if it's a problem with phantomJS (which I know is no longer maintained). But I also tried using the example of rendering a URL to a .png image directly from the **Page Loading** example in the [quickstart](#), and the image came out as just background color with no text, so I'm thinking it's a problem with phantomJS. Or does that example need some extra code to wait for all the included scripts to complete?

Here is the example javascript with the URL in question and image filename hardwired:

```
var page = require('webpage').create();
page.open('https://www.masslottery.com/tools/past-results/mega-millions?
start_date=2021-09-03&end_date=2021-09-28', function(status) {
    console.log("Status: " + status);
    if(status === "success") {
        page.render('example.png');
    }
    phantom.exit();
});
```

Running phantomjs on the above javascript produces this file named example.png as








follows:

But running [foxshot](#) I see the following:

```
$ foxshot 'https://www.masslottery.com/tools/past-results/mega-millions?
start_date=2021-09-03&end_date=2021-09-28'
no valid dimensions provided, defaulting to 1024x768
loading site...
1024x768
```

And it produces a file named screenshot_1024x768.png as follows:


 Games  Tools  Promotions  About  [My Recent Tickets](#)

Home > Tools > Past Results


Past Results



Look up the past results for your favorite Mass Lottery draw games.

Game


Mega Millions 

Date Range View up to 1 year at a time

Sep 3, 2021 - Sep 28, 2021 

8 results  Print Sort: Newest First 

Date	Results	Jackpot
Tue, Sep 28, 2021	18 30 43 68 69 22 4X	\$22 million
Fri, Sep 24, 2021	17 21 27 43 56 15 3X	\$20 million
Tue, Sep 21, 2021	36 41 45 51 56 13 3X	\$432 million
Fri, Sep 17, 2021	17 32 40 59 61 18 3X	\$405 million
Tue, Sep 14, 2021	4 13 19 63 64 16 2X	\$383 million
Fri, Sep 10, 2021	20 32 35 47 64 18 2X	\$368 million

 Help

Of course foxshot doesn't provide a way to access the DOM or otherwise access the text displayed in the image. Any suggestions?

javascript

phantomjs

headless

Share

Improve this question

Follow

edited Nov 27, 2021 at 18:06



Daniel A. White

191k ● 49 ● 377 ● 462

asked Nov 27, 2021 at 17:51



sootsnoot

2,216 ● 3 ● 23 ● 29

Have you tried github.com/puppeteer/puppeteer? It is actively maintained and should be able to render pages to images/pdfs and evaluate javascript on-page. Also if you only need to scrap dates (without rendering images), you don't need a headless browser, you could download HTML with fetch or superagent and then parse it with regexp or github.com/jsdom/jsdom – Afanasii Kurakin Nov 27, 2021 at 18:12 ✎

I tried wget and curl, and neither one of them get the content I'm looking for, Guess I'll try puppeteer. I did update the question to include one of the specific tests I tried to access the DOM with phantomJS. Hope puppeteer can do better. If so, and if you write this comment as an answer, I'll accept the answer. – sootsnoot Nov 28, 2021 at 1:18

1 Answer

Sorted by: Highest score (default) ◆



1



It goes like this with [puppeteer](https://github.com/puppeteer/puppeteer)

```
//npm i puppeteer
const puppeteer = require('puppeteer')

!(async () => {
  //open browser
  const browser = await puppeteer.launch()
  //open browser page
  const page = await browser.newPage()
  //open url
  await page.goto('https://www.masslottery.com/tools/past-results/mega-
millions?start_date=2021-09-03&end_date=2021-09-28')
  //wait for table to appear
  await page.waitForSelector('.multi-col-stacking-table')
  //iterate over table rows on page
  const data = await page.$$eval('.multi-col-stacking-table tbody tr', d =>
    d.map(d => ({
      //format date as YYYY-MM-DD
      date: new Date(d.querySelector('.past-results-row-draw-
date')).innerText().slice(0, 10),
      //collect winning numbers
      winningNumber: Array.from(d.querySelectorAll('.winning-number-ball-
circle, .winning-number-ball-circle-no-border')).map(d => +d.innerText),
      //get multiplier as number
      multiplier: +d.querySelector('.winning-number-ball-
multiplier').innerText().slice(0, -1),
```

```

        //parse jackpot into number - remove all non-digits and multiply by 1M
        jackpot: +d.querySelector('.jackpot').innerText.replace(/[\^d]/g, '') *
1e6
    )))
    )
    console.log(data)
    //optional screenshot
    await page.screenshot({path: 'screenshot.png'})
    await browser.close()
  })()

//output
/*
[
  {
    date: '2021-09-27',
    winningNumber: [ 18, 30, 43, 68, 69, 22 ],
    multiplier: 4,
    jackpot: 22000000
  },
  {
    date: '2021-09-23',
    winningNumber: [ 17, 21, 27, 43, 56, 15 ],
    multiplier: 3,
    jackpot: 20000000
  },
  ...
]
*/

```

Share

edited Nov 29, 2021 at 10:08

answered Nov 28, 2021 at 10:44

Improve this answer



Afanasii Kurakin

3,459 ● 2 ● 27 ● 29

Follow

Wow, thank you so much! I did install nvm/npm/node and then puppeteer, and tried to get something like this to work. But I was unable to get the textContent. I used "const td_elements = await page.\$\$('TABLE.multi-col-stacking-table')", then iterated over td_elements, with "td_elements[i].getProperty('textContent').then(success, failure)". But it always called the failure callback with Protocol Error. – [sootsnoot](#) Nov 28, 2021 at 18:04

Also, the syntax of the 2nd arg to \$\$eval, 'd => d.map(d => { return { totally mystifies me. So I just googled it and see that it's some fancy js shortcut for writing an anonymous function definition - as if javascript needed to be more terse :-). So thanks again!! But the reason I couldn't get my approach to work eludes me. – [sootsnoot](#) Nov 28, 2021 at 18:21

Okay, so after finding out the name of the construct is "arrow functions", I had to google that and found out: "JavaScript arrow functions arrived with the release of ECMAScript 2015, also known as ES6." And what little I know about javascript came around 2013, before such a thing existed. Now I feel slightly less ignorant. – [sootsnoot](#) Nov 28, 2021 at 19:45

1 Glad to help :) – [Afanasii Kurakin](#) Nov 29, 2021 at 10:09