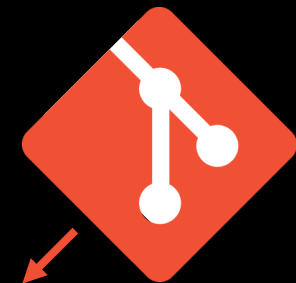
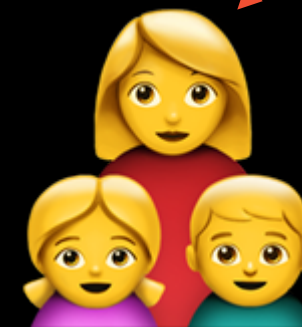


DEVTOOLS & HEADLESS CHROME: THE AUTOMATION POWER-COUPLE

Yonatan Mevorach



@cowchimp

Web Automation



“I can’t automate this”

-- no one ever

(after watching this talk)

Headless Browsers

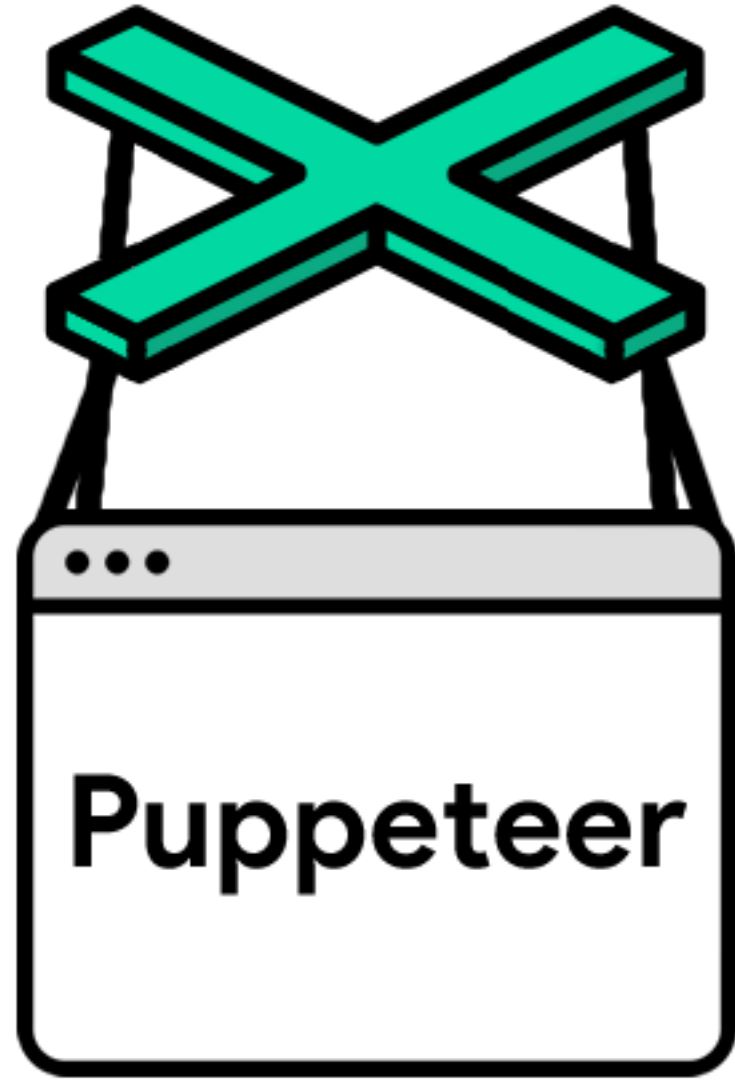




headless



Chrome





find packages



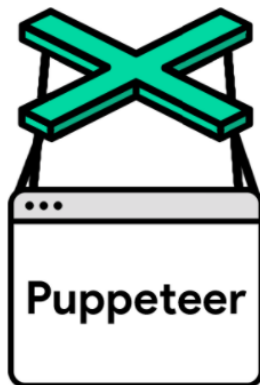
sign up or log in



★ puppeteer public

API | FAQ | Contributing

Puppeteer is a Node library which provides a high-level API to control **headless** Chrome over the **DevTools Protocol**. It can also be configured to use full (non-headless) Chrome.



What can I do?

Most things that you can do manually in the browser can be done using Puppeteer! Here are a few examples to get you started:

- Generate screenshots and PDFs of pages.
- Crawl a SPA and generate pre-rendered content (i.e. "SSR").
- Scrape content from websites.
- Automate form submission, UI testing, keyboard input, etc.
- Create an up-to-date, automated testing environment. Run your tests directly in the latest version of Chrome using the latest JavaScript and browser features.
- Capture a **timeline trace** of your site to help diagnose performance issues.

Getting Started

We ♥ your boss

Selectively mirror the npm registry inside your firewall. Filter packages based on security, licensing, code quality and more. Build awesome stuff faster.

[Try npm Enterprise for free...](#)

📄 `npm install puppeteer`

[how? learn more](#)

 [aslushnikov](#) published 6 days ago

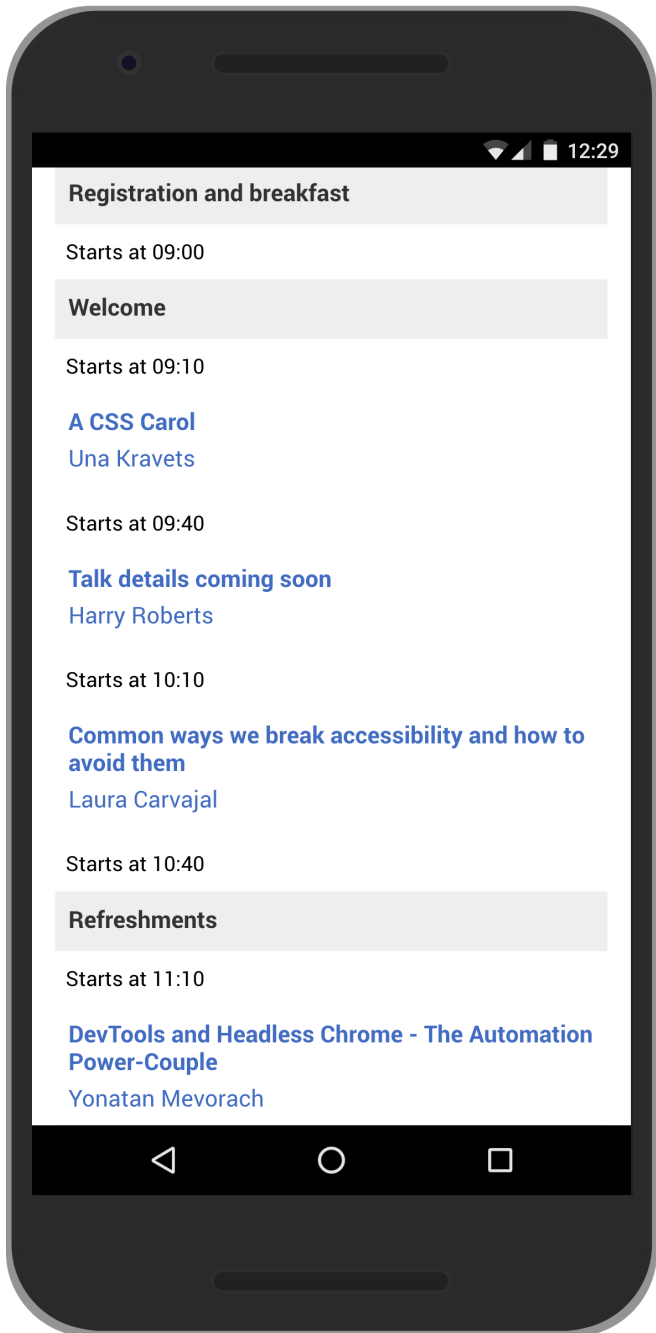
0.11.0 is the latest of 6 releases

github.com/GoogleChrome/puppeteer

Apache-2.0

Collaborators [list](#)





```
[  
  { time: "08:00", title: "Registration and breakfast" },  
  { time: "09:00", title: "Welcome" },  
  { time: "09:10", title: "A CSS Carol" },  
  { time: "09:40", title: "Talk details coming soon" },  
  { time: "10:10", title: "Common ways we break accessibility and how to avoid them" },  
  { time: "10:40", title: "Refreshments" },  
  { time: "11:10", title: "DevTools and Headless Chrome - The Automation Power-Couple" },  
  { time: "11:30", title: "Reducing the friction of online payments" },  
  { time: "11:50", title: "How to hack a web app?" },  
  { time: "12:20", title: "Teach Your Bundler The Users' Habits" },  
  { time: "12:40", title: "Lunch" },  
  { time: "13:40", title: "Creating IoT applications with Web Bluetooth" },  
  { time: "13:55", title: "Escaping the Sandbox with Puck.js" },  
  { time: "14:10", title: "Physical interfaces for digital applications" },  
  { time: "14:25", title: "Let's Get Chatty with Conversational Interface in JavaScript" },  
  { time: "14:55", title: "WebAssembly as cross-platform architecture?!" },  
  { time: "15:25", title: "Refreshments" },  
  { time: "15:55", title: "Data Sketches | A year of exotic data visualizations" },  
  { time: "16:35", title: "Serving for the win - Deploys and hosting for the rest of us" },  
  { time: "17:15", title: "Closing remarks" },  
  { time: "17:25", title: "After Party" }  
]
```

Puppeteer Example

```
1. const puppeteer = require('puppeteer');
2.
3. init();
4.
5. async function init() {
6.   const browser = await puppeteer.launch();
7.
8.   const page = await browser.newPage();
9.
10.  await page.goto('https://2018.render-conf.com/schedule');
11.
12.  const schedule = await page.evaluate(() => {
13.    return Array.from(
```



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

```
2.
```

```
3. init();
```

```
4.
```

require Puppeteer

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

```
6.
```

```
1. const puppeteer = require('puppeteer');
2.
3. init();
4.
5. async function init() {
6.     const browser = await puppeteer.launch();
7.
8.     const page = await browser.newPage();
```

launch Chrome

```
1. const puppeteer = require('puppeteer');
2.
3. init();
4.
5. async function init() {
6.   const browser = await puppeteer.launch();
7.
8.   const page = await browser.newPage();
9.
10.  await page.goto('https://2018.render-conf.com/schedule');
11.
```

open a new tab

```
13.   return Array.from(
14.     await page.evaluate(() => {
```



```
1. const puppeteer = require('puppeteer');
2.
3. init();
4.
5. async function init() {
6.   const browser = await puppeteer.launch();
7.
8.   const page = await browser.newPage();
9.
10.  await page.goto('https://2018.render-conf.com/schedule');
11.
12.  const schedule = await page.evaluate(() => {
13.    return Array.from(
```

go to Render Conf schedule page

```
1. const puppeteer = require('puppeteer');
2.
3. init();
4.
5. async function init() {
6.   const browser = await puppeteer.launch();
7.
8.   const page = await browser.newPage();
9.
10.  await page.goto('https://2018.render-conf.com/schedule');
11.
12.  const schedule = await page.evaluate(() => {
13.    return Array.from(
14.      document.querySelectorAll('.slot')
15.    )
```

execute JS on the client

```
17.     title: x.querySelector('.session__title').innerText
```

```
18.   )
```

```
1.
2.
3.  init();
4.
5.  async function init() {
6.    const browser = await puppeteer.launch();
7.
8.    const page = await browser.newPage();
9.
10.   await page.goto('https://2018.render-conf.com/schedule');
11.
12.   const schedule = await page.evaluate(() => {
13.     return Array.from(
14.       document.querySelectorAll('.slot')
15.     ).map(x => ({
16.       time: x.querySelector('.session-start-time').innerText,
17.       title: x.querySelector('.session__title').innerText
18.     }));
19.   });
```

extract schedule by querying the DOM

```
21.   console.log(schedule);
```

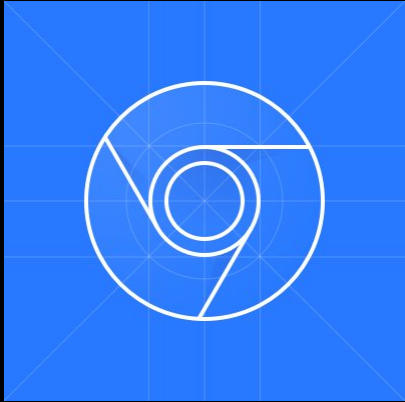


```
8.   const page = await browser.newPage();
9.
10.  await page.goto('https://2018.render-conf.com/schedule');
11.
12.  const schedule = await page.evaluate(() => {
13.    return Array.from(
14.      document.querySelectorAll('.slot')
15.    ).map(x => ({
16.      time: x.querySelector('.session-start-time').innerText,
17.      title: x.querySelector('.session__title').innerText
18.    }));
19.  });
20.
21.  console.log(schedule);
22.
23.  await browser.close();
```

print the result

```
10.   await page.goto('https://2018.render-conf.com/schedule');
11.
12.   const schedule = await page.evaluate(() => {
13.     return Array.from(
14.       document.querySelectorAll('.slot')
15.     ).map(x => ({
16.       time: x.querySelector('.session-start-time').innerText,
17.       title: x.querySelector('.session__title').innerText
18.     }));
19.   });
20.
21.   console.log(schedule);
22.
23.   await browser.close();
24. }
```

close the browser



Chrome DevTools



h3.hero__subheading | 503.53 x 108

Render Conf

23 MARCH 2018 // OXFORD, UK

Elements Console Sources Network Performance Memory Application Security Audits

```
<div class="col-sm-10 col-sm-offset-1">  
  <div class="hero__content">  
    <h3 class="hero__subheading">Render Conf</h3> == $0  
    <ul class="hero__text">...</ul>  
    <div class="text-center">...</div>  
  </div>  
  ::after  
</div>  
  ::after  
</div>  
</div>  
<div id="confetti-container" class="confetti-container">...</div>  
</section>  
<section class="section">...</section>
```

html body section div div div div.col-sm-10.col-sm-offset-1 div.hero__content h3.hero__subheading

Styles Computed >>

Filter :hov .cls +

```
ng {  
  font-size: 5em;  
}  
  
@media screen and (min-width: 992px)  
.hero__subhead main.min.css:8  
ng {  
  font-size: 4em;  
  max-width: none;  
}  
  
@media screen and (min-width: 768px)
```



h3.hero_subheading | 503.53 x 108

Render Conf

23 MARCH 2018 // OXFORD, UK

Elements Console Sources Network Performance Memory Application Security Audits

```
<div class="col-sm-10 col-sm-offset-1">  
  <div class="hero_content">  
    <h3 class="hero_subheading">Render Conf</h3> == $0  
    <ul class="hero_text">...</ul>  
    <div class="text-center">...</div>  
  </div>  
</div>
```

Styles Computed >>

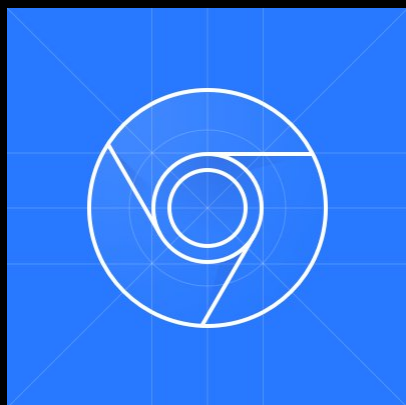
Filter :hov .cls +

```
ng {  
  font-size: 5em;
```

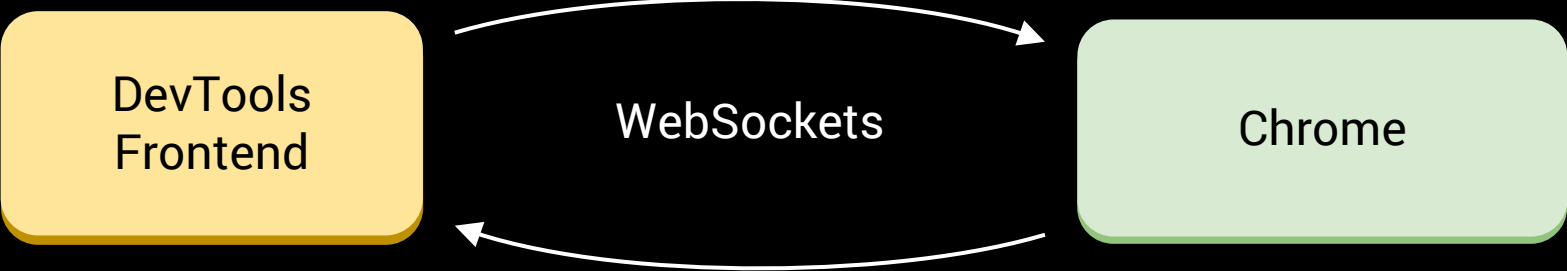
Console Coverage Changes x Network conditions Rendering

```
main.min.css  
4 4 */@font-face{font-family:FontAwesome;src:url(../fonts/fontawesome-webfont.eot?v=4.6.3);src:url(../font  
5 5 * Bootstrap v3.3.6 (http://getbootstrap.com)  
6 6 * Copyright 2011-2015 Twitter, Inc.  
7 7 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)  
8 *///! normalize.css v3.0.3 | MIT License | github.com/necolas/normalize.css */html{font-family:sans-ser  
8 *///! normalize.css v3.0.3 | MIT License | github.com/necolas/normalize.css */html{font-family:sans-ser
```

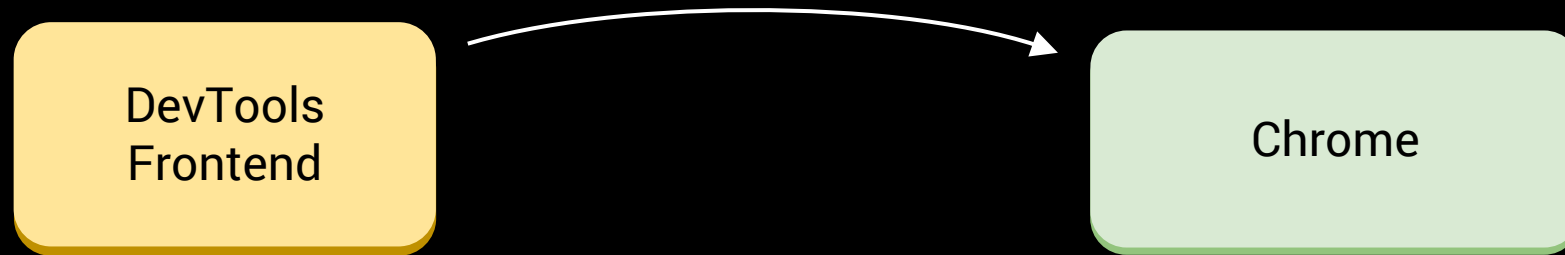
1 insertion (+), 1 deletion (-)



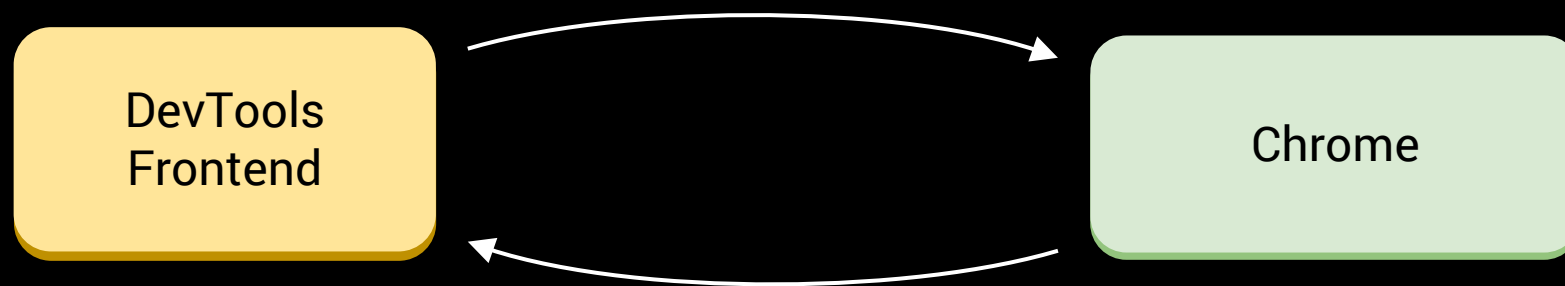
protocol
Chrome DevTools



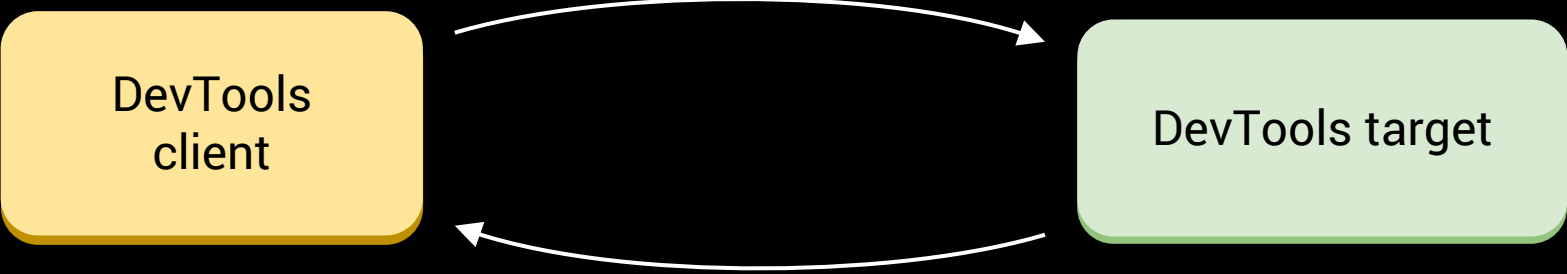
```
{  
  id: 1,  
  method: "Runtime.evaluate",  
  params: { expression: "1 + 1" }  
}
```



```
{
  id: 1,
  method: "Runtime.evaluate",
  params: { expression: "1 + 1" }
}
```



```
{
  id: 1,
  result: {
    result: { type: "number", value: 2, description: "2" },
    wasThrown: false
  }
}
```




```
node --inspect index.js
```

C:\Users\Yonata...-master\app.js x

```
1 (function (exports, require, module, __filename, __dirname) { var http = require('http');
2   path = require('path'),
3   methods = require('methods'),
4   express = require('express'),
5   bodyParser = require('body-parser'),
6   session = require('express-session'),
7   cors = require('cors'),
8   passport = require('passport'),
9   errorHandler = require('errorhandler'),
10  mongoose = require('mongoose');
11
12  var isProduction = process.env.NODE_ENV === 'production';
13
14  // Create global app object
15  var app = express();
16
17  app.use(cors());
18
19  // Normal express config defaults
20  app.use(require('morgan')('dev'));
21  app.use(bodyParser.urlencoded({ extended: false }));
22  app.use(bodyParser.json());
23
24  app.use(require('method-override')());
25  app.use(express.static(__dirname + '/public'));
26
27
```

**Debugger paused**

▼ Threads

▶ Node.js: file:///C:/_Users_Yonatan_...
paused

▶ Watch

▼ Call Stack

▶ (anonymous)
C:\Users\Yonata...aster\app.js:1

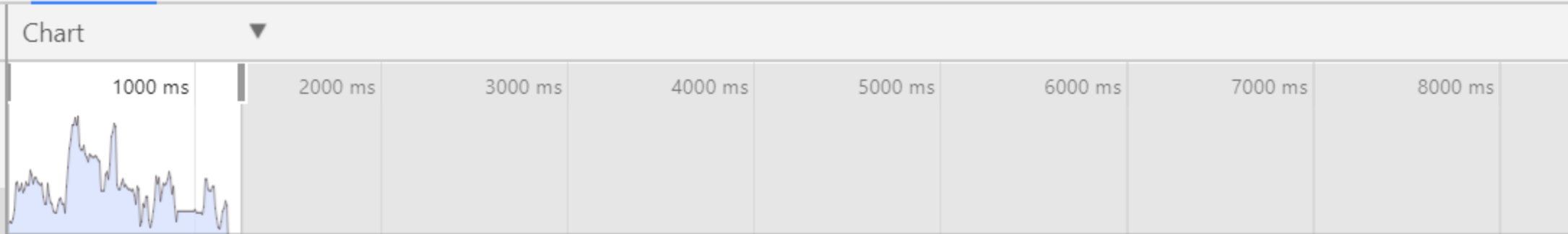
Module._compile	module.js:566
Module._extensions.js	module.js:580
Module.load	module.js:503
tryModuleLoad	module.js:466
Module._load	module.js:458
Module.runMain	module.js:605
startup	bootstrap_node.js:158
(anonymous)	bootstrap_node.js:575

▼ Scope

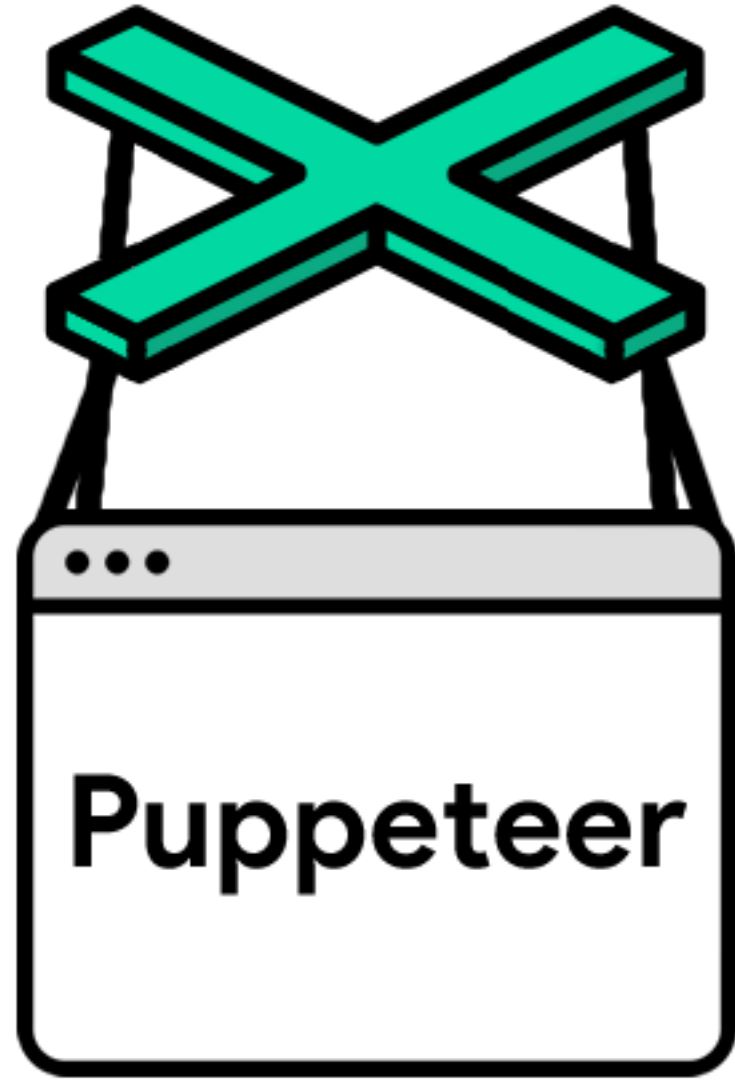
Profiles

CPU PROFILES

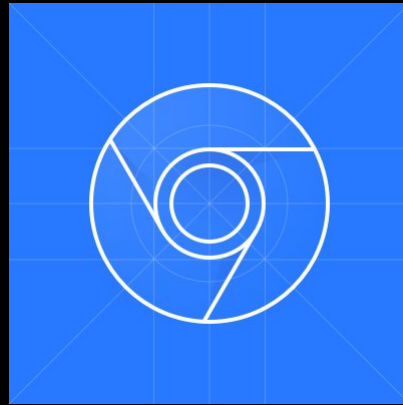
Profile 1 [Save](#)



Function	Start Time	End Time	Duration
(anonymous)	0 ms	200 ms	200 ms
require	200 ms	400 ms	200 ms
Module.require	200 ms	400 ms	200 ms
Module.load	200 ms	400 ms	200 ms
tryModuleLoad	200 ms	400 ms	200 ms
Module.load	200 ms	400 ms	200 ms
Module_extensions.js	200 ms	400 ms	200 ms
Module_compile	200 ms	400 ms	200 ms
(anonymous)	400 ms	600 ms	200 ms
require	400 ms	600 ms	200 ms
Module.require	400 ms	600 ms	200 ms
Module.load	400 ms	600 ms	200 ms
tryModuleLoad	400 ms	600 ms	200 ms
Module.load	400 ms	600 ms	200 ms
Module_extensions.js	400 ms	600 ms	200 ms
Module_compile	400 ms	600 ms	200 ms
(anonymous)	600 ms	800 ms	200 ms
require	600 ms	800 ms	200 ms
Module.require	600 ms	800 ms	200 ms
Module.load	600 ms	800 ms	200 ms
tryModuleLoad	600 ms	800 ms	200 ms
Module.load	600 ms	800 ms	200 ms
Module_extensions.js	600 ms	800 ms	200 ms
Module_compile	600 ms	800 ms	200 ms
(anonymous)	800 ms	1000 ms	200 ms
require	800 ms	1000 ms	200 ms
Module.require	800 ms	1000 ms	200 ms
Module.load	800 ms	1000 ms	200 ms
tryModuleLoad	800 ms	1000 ms	200 ms
Module.load	800 ms	1000 ms	200 ms
Module_extensions.js	800 ms	1000 ms	200 ms
Module_compile	800 ms	1000 ms	200 ms
(anonymous)	1000 ms	1200 ms	200 ms
require	1000 ms	1200 ms	200 ms
Module.require	1000 ms	1200 ms	200 ms
Module.load	1000 ms	1200 ms	200 ms
tryModuleLoad	1000 ms	1200 ms	200 ms
Module.load	1000 ms	1200 ms	200 ms
Module_extensions.js	1000 ms	1200 ms	200 ms
Module_compile	1000 ms	1200 ms	200 ms



headless



protocol

chrome

--headless

--remote-debugging-port=9222

<https://2018.render-conf.com>

chrome

--headless

--remote-debugging-port=9222

<https://2018.render-conf.com>

chrome

--headless

--remote-debugging-port=9222

https://2018.render-conf.com

chrome

--headless

--remote-debugging-port=9222

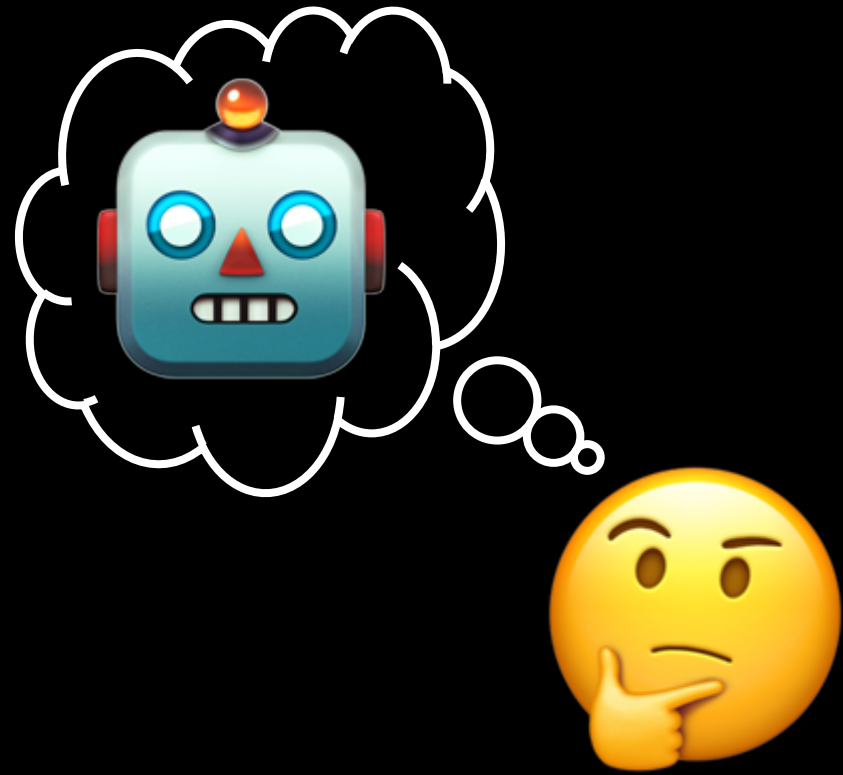
https://2018.render-conf.com

chrome

--headless

--remote-debugging-port=9222

<https://2018.render-conf.com>



Unused CSS Reporter

```
1. const browser = await puppeteer.launch();
2.
3. const page = await browser.newPage();
4.
5. const c = page._client;
6. await c.send('DOM.enable');
7. await c.send('CSS.enable');
8.
9. await c.send('CSS.startRuleUsageTracking');
10.
11. const stylesheets = [];
12.   c.on('CSS.styleSheetAdded', stylesheet => {
13.     stylesheets.push(stylesheet.header);
```

```
1. const browser = await puppeteer.launch();  
2.  
3. const page = await browser.newPage();  
4.
```

launch Chrome with Puppeteer and open a new tab

```
7. await c.send( 'css.enable' );  
8.
```



```
1. const browser = await puppeteer.launch();
2.
3. const page = await browser.newPage();
4.
5. const c = page._client;
6. await c.send('DOM.enable');
7. await c.send('CSS.enable');
8.
9. await c.send('CSS.startRuleUsageTracking');
10.
11. const stylesheets = [];
12. page.on('css:stylesheetAdded', (stylesheet) => {
```

start tracking CSS usage

```
14. },
```

```
15.
```

```
1. const browser = await puppeteer.launch();
2.
3. const page = await browser.newPage();
4.
5. const c = page._client;
6. await c.send('DOM.enable');
7. await c.send('CSS.enable');
8.
9. await c.send('CSS.startRuleUsageTracking');
10.
11. const stylesheets = [];
12.   c.on('CSS.styleSheetAdded', stylesheet => {
13.     stylesheets.push(stylesheet.header);
14.   });
15.
```

listen for when a stylesheet is loaded

```
18. await page.click('.tab.type2');
```

```
5. const c = page._client;
6. await c.send('DOM.enable');
7. await c.send('CSS.enable');
8.
9. await c.send('CSS.startRuleUsageTracking');
10.
11. const stylesheets = [];
12.   c.on('CSS.styleSheetAdded', stylesheet => {
13.     stylesheets.push(stylesheet.header);
14.   });
15.
16. const url = 'https://example-site-izvafmtnha.now.sh/';
17. await page.goto(url);
18. await page.click('.tab.type2');
19. await page.click('.tab.type3');
20. await page.click('.tab.type4');
```

interact with the page (load the page and click tabs)

```
9. await c.send('CSS.startRuleUsageTracking');
10.
11. const stylesheets = [];
12.   c.on('CSS.styleSheetAdded', stylesheet => {
13.     stylesheets.push(stylesheet.header);
14.   });
15.
16. const url = 'https://example-site-izvafmtnha.now.sh/';
17. await page.goto(url);
18. await page.click('.tab.type2');
19. await page.click('.tab.type3');
20. await page.click('.tab.type4');
21.
22. const {ruleUsage} = await c.send('CSS.stopRuleUsageTracking');
23.
24. const unusedCSS = calcUnusedCss(ruleUsage, stylesheets);
25.
```

stop tracking CSS usage and fetch used rules

```
11. const stylesheets = [];  
12.   c.on('CSS.styleSheetAdded', stylesheet => {  
13.     stylesheets.push(stylesheet.header);  
14.   });  
15.  
16. const url = 'https://example-site-izvafmtnha.now.sh/';  
17. await page.goto(url);  
18. await page.click('.tab.type2');  
19. await page.click('.tab.type3');  
20. await page.click('.tab.type4');  
21.  
22. const {ruleUsage} = await c.send('CSS.stopRuleUsageTracking');  
23.  
24. const unusedCSS = calcUnusedCss(ruleUsage, stylesheets);  
25.  
26. console.log(`${unusedCSS}% of your CSS is unused`);  
27.
```

crunch the numbers

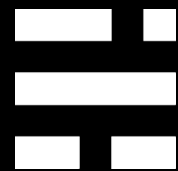
```
14. });
15.
16. const url = 'https://example-site-izvafmtnha.now.sh/';
17. await page.goto(url);
18. await page.click('.tab.type2');
19. await page.click('.tab.type3');
20. await page.click('.tab.type4');
21.
22. const {ruleUsage} = await c.send('CSS.stopRuleUsageTracking');
23.
24. const unusedCSS = calcUnusedCss(ruleUsage, stylesheets);
25.
26. console.log(`${unusedCSS}% of your CSS is unused`);
27.
28. await browser.close();
29.
```

print the result and close the browser

**tinyurl.com/
headless-talk**



@cowchimp



blog.cowchimp.com