



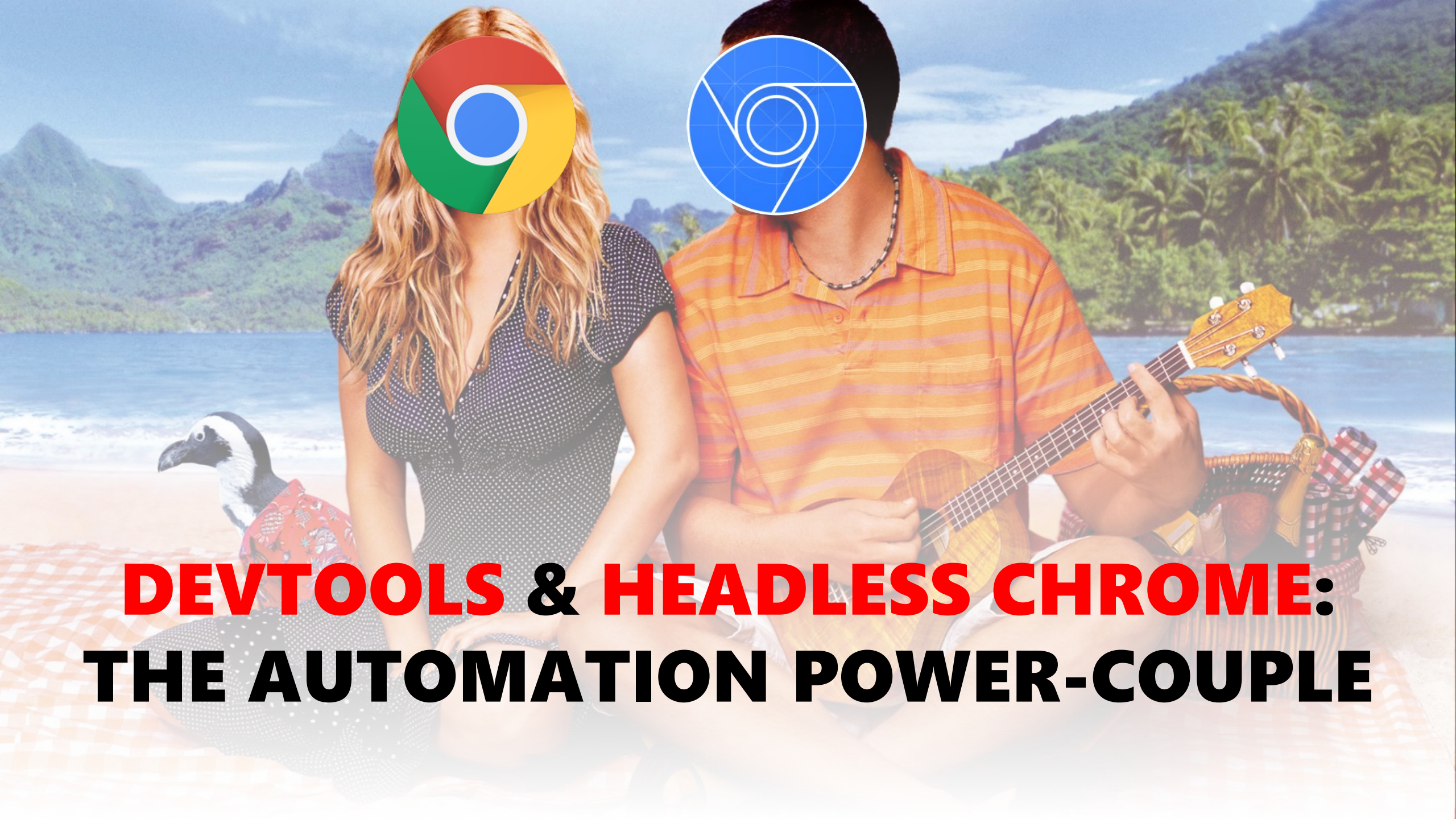
Yonatan Mevorach

Web Developer

wix.com

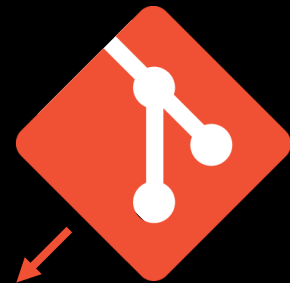


FRONTCON



DEVTOOLS & HEADLESS CHROME: THE AUTOMATION POWER-COUPLE

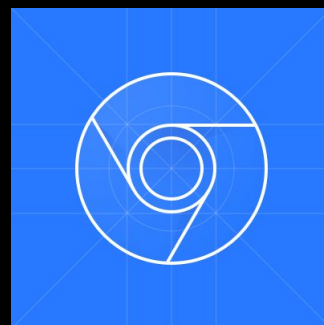
Yonatan Mevorach



@cowchimp



WiXEngineering



Web Automation



“I can’t automate this”

-- no one ever

(after watching this talk)

Headless Browsers

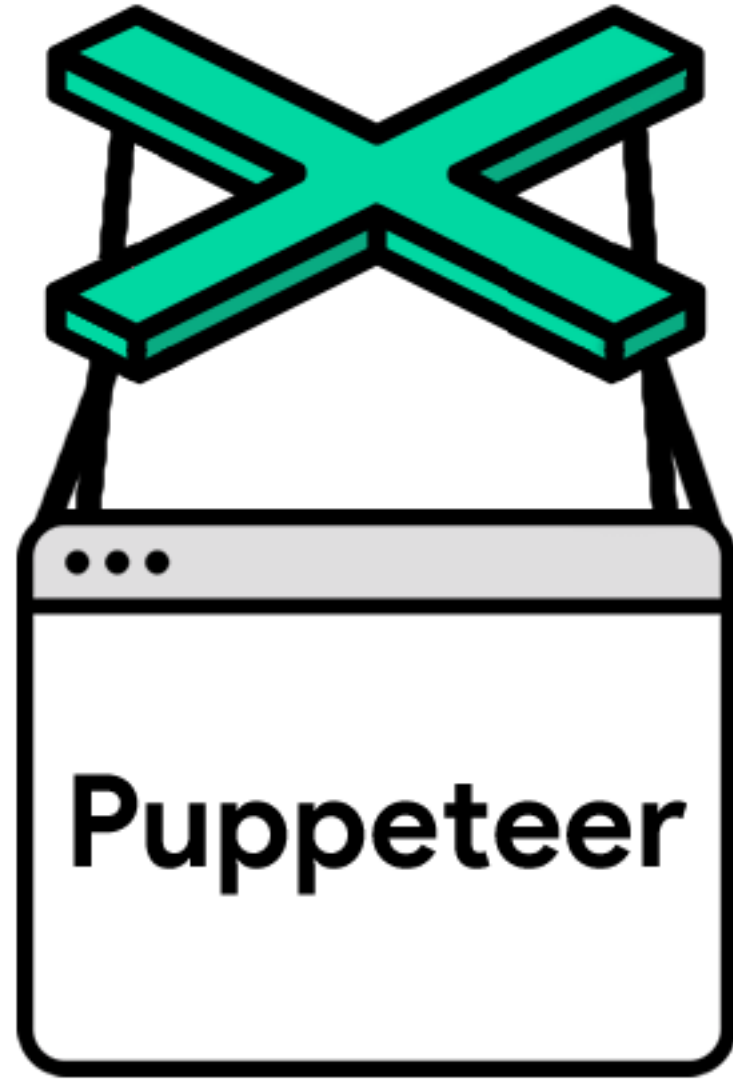




headless



Chrome





find packages



sign up or log in



★ puppeteer public

API | FAQ | Contributing

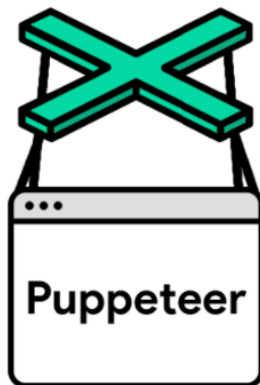
Puppeteer is a Node library which provides a high-level API to control **headless** Chrome over the **DevTools Protocol**. It can also be configured to use full (non-headless) Chrome.

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

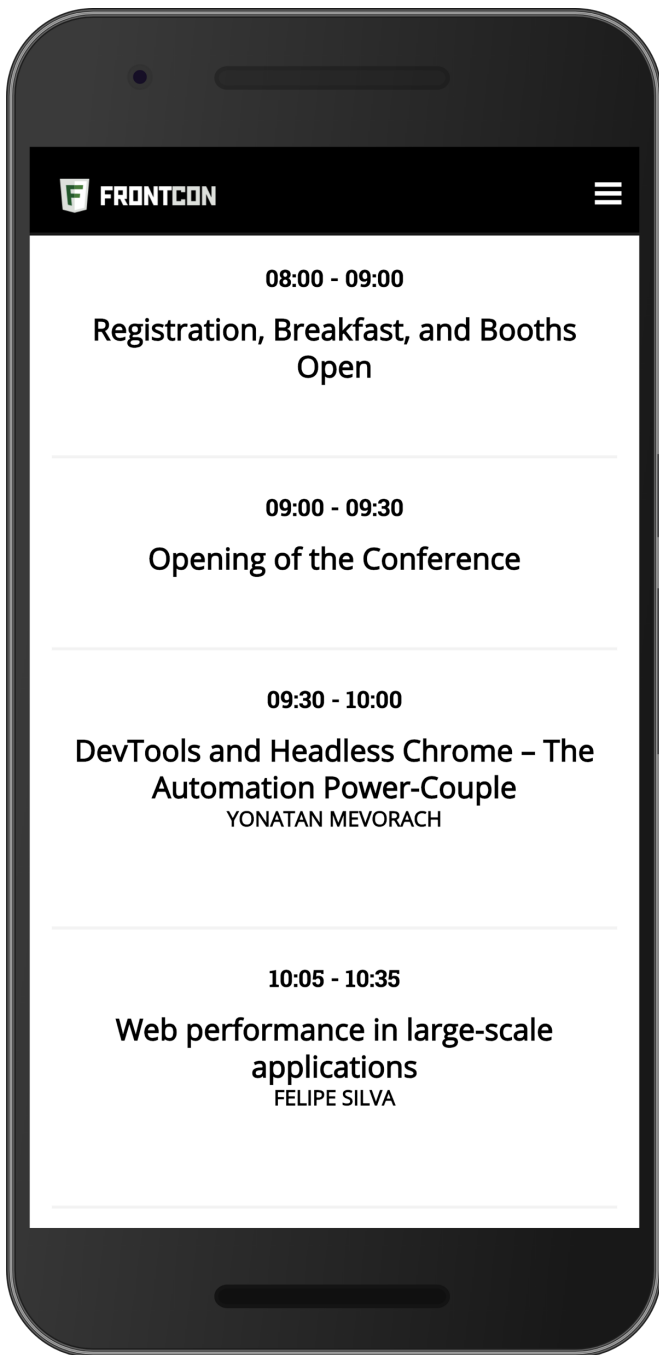
0.11.0 is the latest of 6 releases

github.com/GoogleChrome/puppeteer

Apache-2.0

Collaborators [list](#)





```
[
  { "time": "08:00 - 09:00", "title": "Registration, Breakfast, and Booths Open" },
  { "time": "09:00 - 09:30", "title": "Opening of the Conference" },
  { "time": "09:30 - 10:00", "title": "DevTools and Headless Chrome - The Automation Power-Couple" },
  { "time": "10:05 - 10:35", "title": "Web performance in large-scale applications" },
  { "time": "10:40 - 11:10", "title": "Backends for frontends with GraphQL" },
  { "time": "11:10 - 11:45", "title": "Break" },
  { "time": "11:45 - 12:15", "title": "JavaScript security: a retrospective" },
  { "time": "12:20 - 12:50", "title": "Moving Web applications into Virtual Reality" },
  { "time": "12:55 - 13:50", "title": "Lunch" },
  { "time": "13:55 - 14:20", "title": "OpenSpace opening" },
  { "time": "14:25 - 14:55", "title": "Channel your inner rockstar with the web audio API." },
  { "time": "15:00 - 15:25", "title": "Break" },
  { "time": "15:30 - 16:00", "title": "Challenge Accepted: Scoping Errors in “Multi-App” Environment" },
  { "time": "16:05 - 16:40", "title": "Getting rid of runtime errors with Elm" },
  { "time": "16:40 - 17:00", "title": "Close Day" },
  { "time": "19:00 - ...", "title": "Dinner and Party" }
]
```



16:01
JUN 08

fitbit

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://frontcon.lv/schedule', {
11.        waitUntil: 'networkidle0'
12.    });
13.
```

```
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();
9.
```

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://frontcon.lv/schedule', {
11.    waitUntil: 'networkidle0'
```

open a new tab

```
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://frontcon.lv/schedule', {
11.        waitUntil: 'networkidle0'
12.    });
13.
14.    const schedule = await page.evaluate(() => {
```

go to FrontCon schedule page

SCHEDULE



Elements

Console

Sources



top



Filter

Default levels



Grc



```
> Array.from(
  document.querySelectorAll('.schedule__item')
).map(x => ({
  time: x.querySelector('.schedule__time').innerText,
  title: x.querySelector('.schedule__event').innerText
}))
```



```
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://frontcon.lv/schedule', {
11.    waitUntil: 'networkidle0'
12.  });
13.
14.  const schedule = await page.evaluate(() => {
15.    return Array.from(
16.      document.querySelectorAll('.schedule__item')
17.    );
18.  });
19.
20.  return {
21.    title: x.querySelector('.schedule__event').innerText
22.  };
23. }
```

execute JS on the client

```
4.
5. async function init() {
6.     const browser = await puppeteer.launch();
7.
8.     const page = await browser.newPage();
9.
10.    await page.goto('https://frontcon.lv/schedule', {
11.        waitUntil: 'networkidle0'
12.    });
13.
14.    const schedule = await page.evaluate(() => {
15.        return Array.from(
16.            document.querySelectorAll('.schedule__item')
17.        ).map(x => ({
18.            time: x.querySelector('.schedule__time').innerText,
19.            title: x.querySelector('.schedule__event').innerText
20.        }));
21.    });
22.
23.    // console.log(schedule);
24.}
```

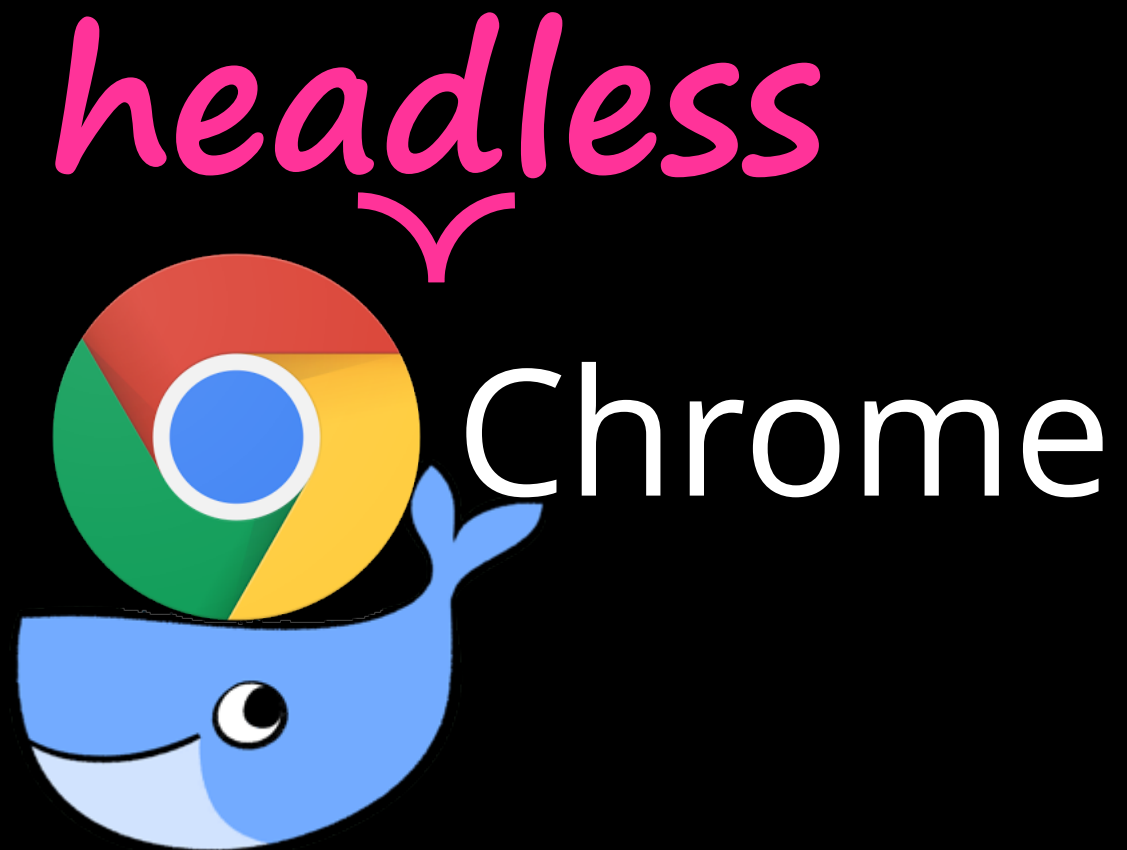
extract schedule by querying the DOM

```
10.     await page.goto('https://frontcon.lv/schedule', {
11.         waitUntil: 'networkidle0'
12.     });
13.
14.     const schedule = await page.evaluate(() => {
15.         return Array.from(
16.             document.querySelectorAll('.schedule__item')
17.         ).map(x => ({
18.             time: x.querySelector('.schedule__time').innerText,
19.             title: x.querySelector('.schedule__event').innerText
20.         }));
21.     });
22.
23.     console.log(schedule);
24.
25.     await browser.close();
26. }
```

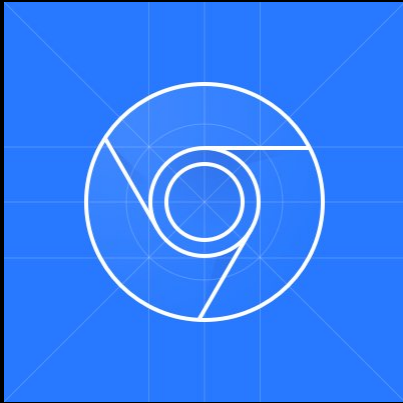
print the result

```
12.     });
13.
14.     const schedule = await page.evaluate(() => {
15.     return Array.from(
16.         document.querySelectorAll('.schedule__item')
17.     ).map(x => ({
18.         time: x.querySelector('.schedule__time').innerText,
19.         title: x.querySelector('.schedule__event').innerText
20.     }));
21. });
22.
23.     console.log(schedule);
24.
25.     await browser.close();
26. }
```

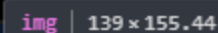
close the browser







Chrome DevTools




FRONT-END CONFERENCE

June 14-15, 2018 Riga, Latvia

html body div div #efcb-conference-288 div div div.time-schedule div.time-schedule_pic img

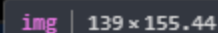
Styles Computed >>

Filter :hov .cls +

```
.time-  
schedule__pic img {  
  display: block;  
  width: 100%;  
  height: auto;  
  position: static;  
}
```

html, main.css?ver=4.9.6:5

body, div, span, h1, h2, h3,
h4, h5, h6, ...




FRONT-END CONFERENCE




[Elements](#)
[Console](#)
[Sources](#)
[Network](#)
[Performance](#)
[Memory](#)
[Application](#)
[Security](#)
[Audits](#)

```

<!-- time-schedule__pic -->
<div class="time-schedule__pic">
   == $0
</div>
<!-- /time-schedule__pic -->

```

html

body

div

div

#efcb-conference-288

div

div

div.time-schedule

div.time-schedule__pic

img

Styles

Computed

>>

Filter

:hov .cls +

+

.time- main.css?ver=4.9.6:1129

schedule__pic img {

display: block;

Console

Coverage

Changes x

Network conditions

Rendering

What's New

main.css?ver=4.9.6

1132 1132

height: auto;

1133

position: static;

1133 1134

}

1 insertion (+), 0 deletions (-)



ChromeDevTools / devtools-frontend

Watch

41

★ Star

500

🔗 Fork

65

<> Code

🔔 Issues 10

🔗 Pull requests 1

📁 Projects 0

📖 Wiki

Insights ▼

The Chrome DevTools UI <https://www.npmjs.com/package/chrome-...>

chrome-devtools

javascript

instrumentation

🔄 3,598 commits

🔗 1 branch

📦 0 releases

👤 102 contributors

Branch: master ▼

New pull request

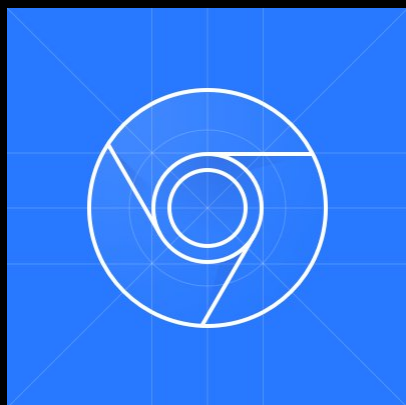
Find file

Clone or download ▼

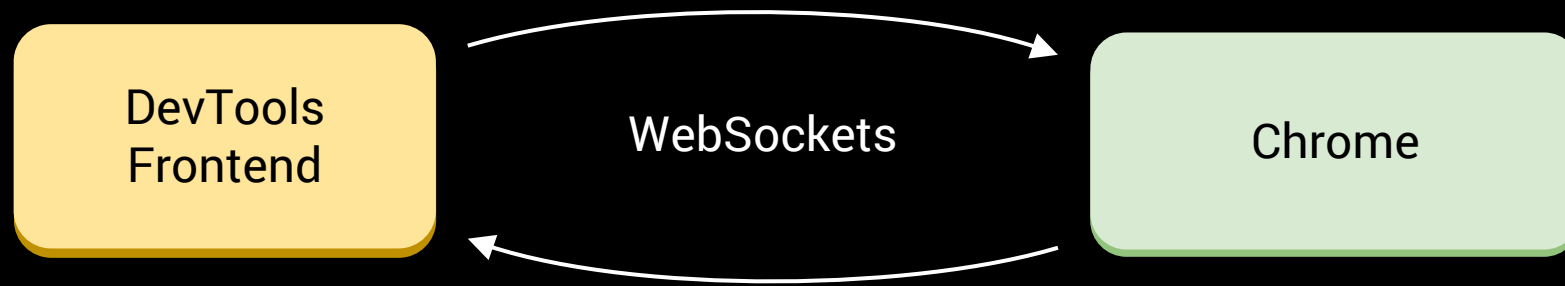
👤 eugeneo committed with Commit Bot [DevTools] Allow the user to customize SW events ...

Latest commit 0394cbf 4 days ago

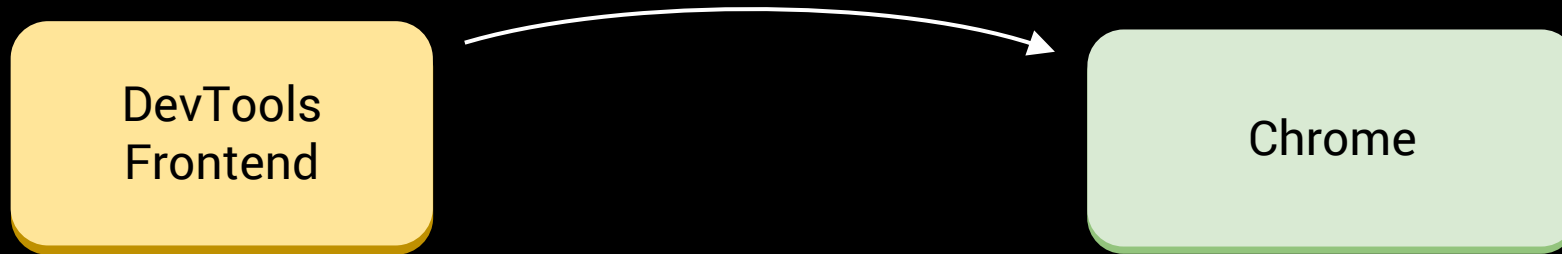
📁 front_end	[DevTools] Allow the user to customize SW events	4 days ago
📁 scripts	DevTools: move http/tests/inspector-enabled tests (part 13)	5 days ago
📁 services	[DevTools] Move main target/connection to TargetManager.	11 months ago
📄 .clang-format	DevTools: clang format all the source + clang config landed.	11 months ago
📄 .eslintignore	DevTools: Validate module.json	3 months ago
📄 .eslintrc.js	DevTools: use contexts per domain when logging protocol messages	a month ago
📄 .gitignore	DevTools: Allow tabbing into styles sidebar	a month ago
📄 .npmignore	DevTools: Added package.json, LICENSE, and .npmignore	a year ago
📄 .style.yapf	DevTools: format python scripts using yapf	8 months ago



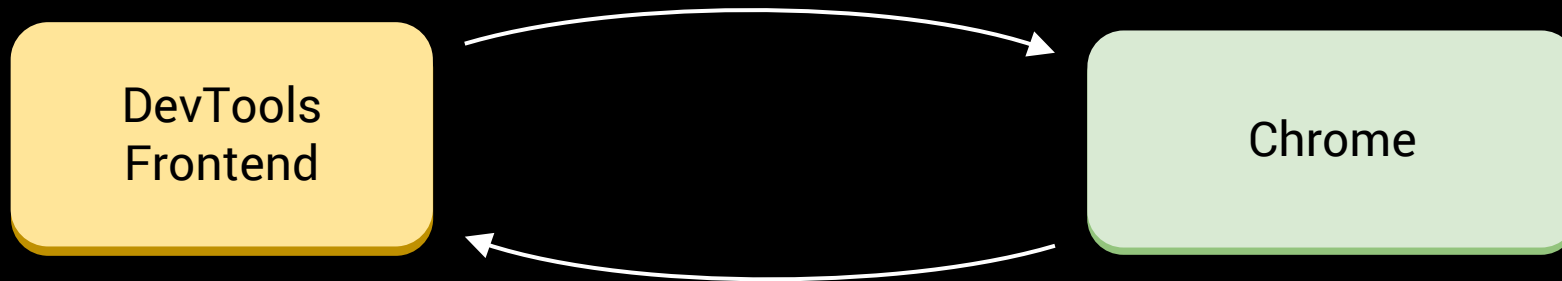
protocol
Chrome DevTools



```
{  
  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  
  }  
}
```

Domains

Accessibility

Animation

ApplicationCache

Audits

Browser

CSS

CacheStorage

Console

DOM

DOMDebugger

DOMSnapshot

DOMStorage

The **Chrome DevTools Protocol** allows for tools to instrument, inspect, debug and profile Chromium, Chrome and other Blink-based browsers. Many existing projects [currently use](#) the protocol. The [Chrome DevTools](#) uses this protocol and the team maintains its API.

Instrumentation is divided into a number of domains (DOM, Debugger, Network etc.). Each domain defines a number of commands it supports and events it generates. Both commands and events are serialized JSON objects of a fixed structure. You can either [debug over the wire](#) using the raw messages as they are described in the corresponding domain documentation, or use [extension JavaScript API](#).

Protocol API Docs

[The latest \(tip-of-tree\) protocol \(tot\)](#)

It [changes](#) frequently and can break at any time. However it captures the full capabilities of the Protocol, whereas the stable release is a subset. There is no backwards compatibility support guaranteed for the capabilities it introduces.

[v8-inspector protocol \(v8\)](#)

It is available in [node 6.3+](#) and enables [debugging & profiling](#) of Node.js apps.

[stable 1.2 protocol \(1-2\)](#)

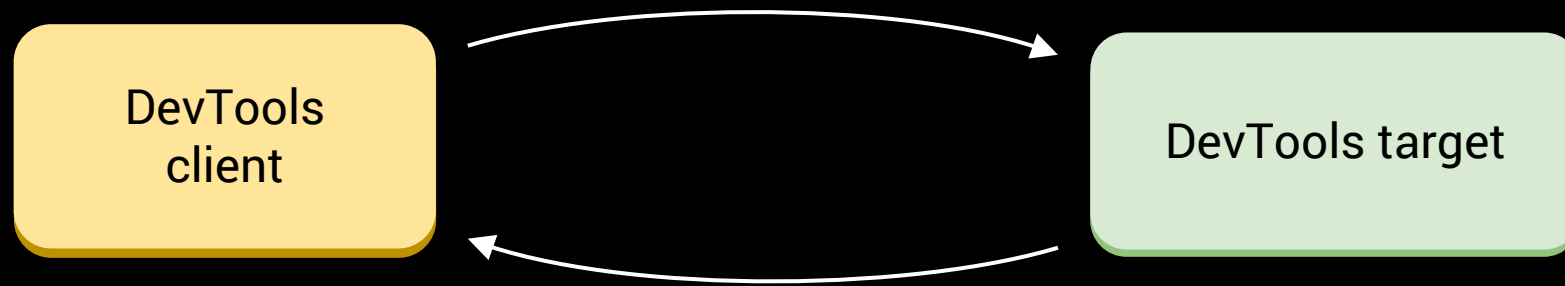
The stable release of the protocol, tagged at Chrome 54. It includes a smaller subset of the complete protocol compatibilities.

Resources

Consider subscribing to the [chrome-debugging-protocol](#) mailing list.

The [devtools-protocol repo](#) issue tracker can also be used for concerns with the protocol. It also hosts the canonical copy of the json files.

Fork me on GitHub



```
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

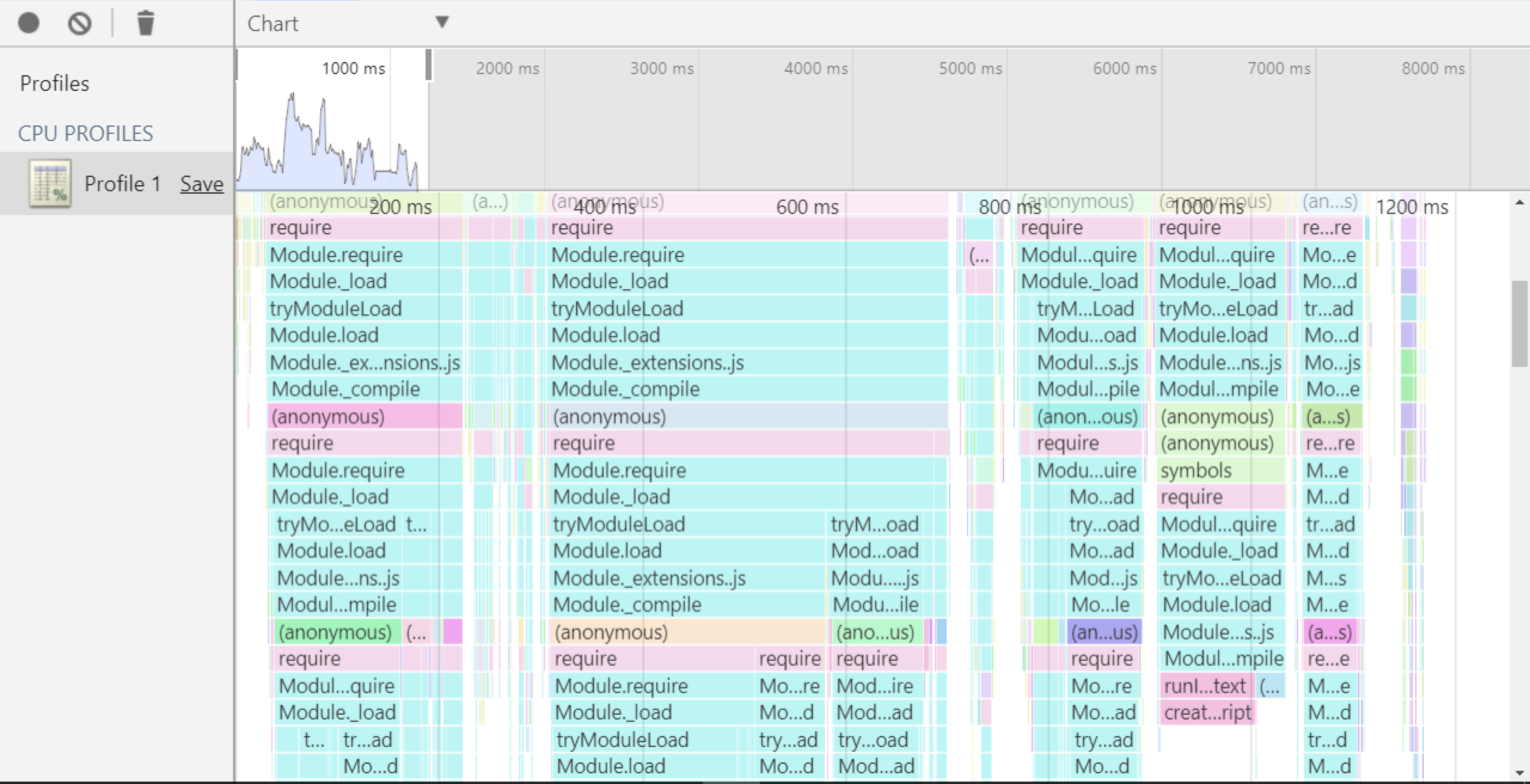
Node.js: file:///C:/_Users_Yonatan_...
paused

Watch

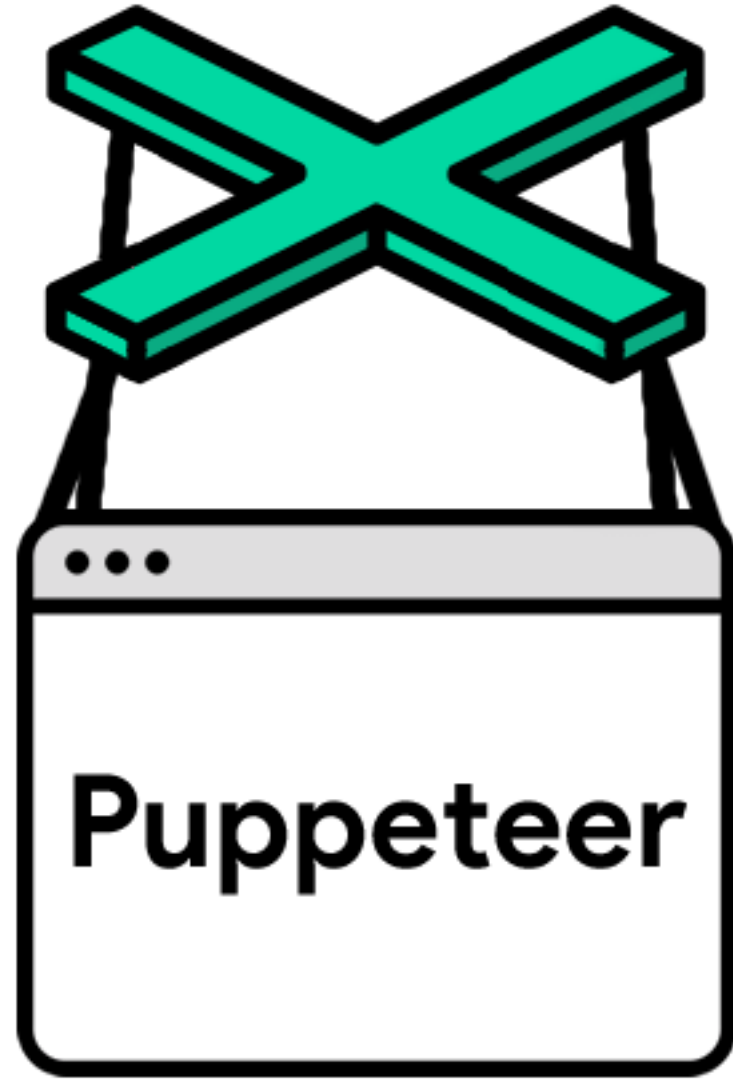
Call Stack

(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







headless    *protocol*

chrome

--headless

--remote-debugging-port=9222

<https://frontcon.lv>

chrome

--headless

--remote-debugging-port=9222

<https://frontcon.lv>

chrome

--headless

--remote-debugging-port=9222

https://frontcon.lv

chrome

--headless

--remote-debugging-port=9222

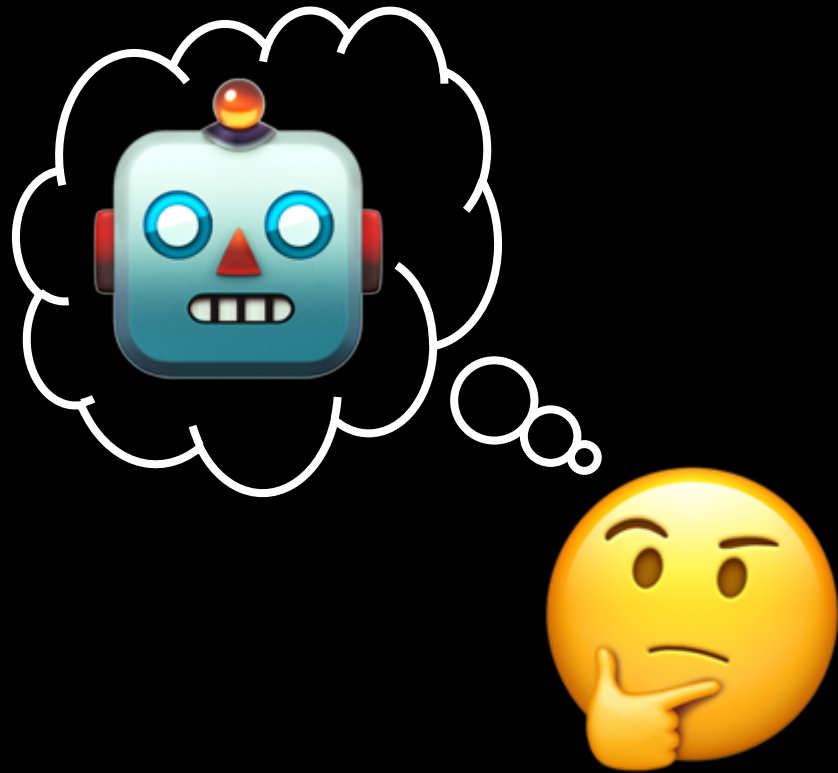
https://frontcon.lv

chrome

--headless

--remote-debugging-port=9222

<https://frontcon.lv>



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

```
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.
```

enable necessary "Domains" using raw DevTools Protocol commands

```
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

```
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-izvafmtinha.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-izvafmtinha.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.
```

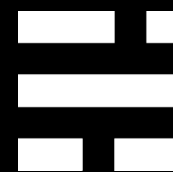
print the result and close the browser

Recap!



**tinyurl.com/
headless-talk**

 **@cowchimp**

 **blog.cowchimp.com**