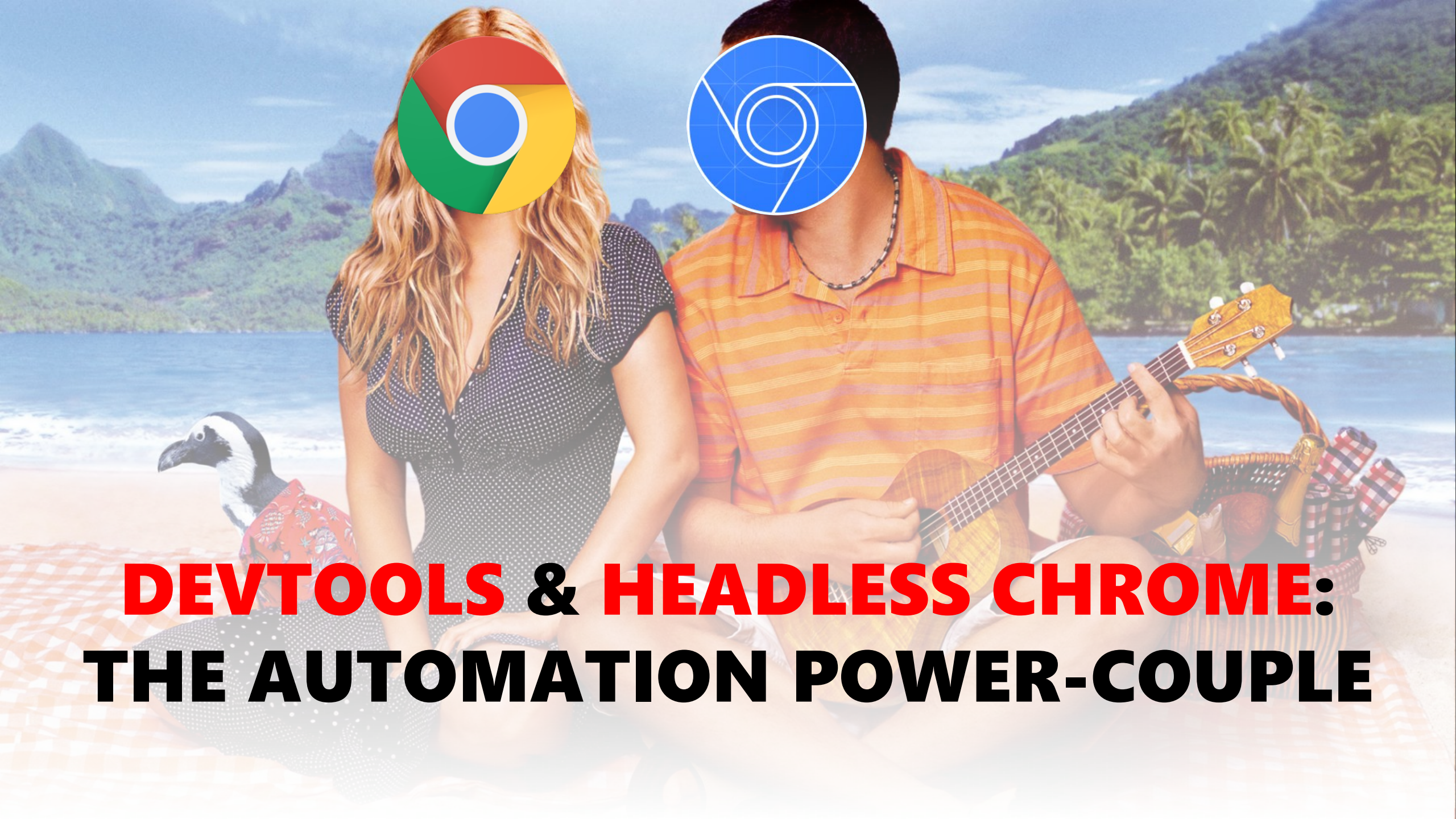




DEVTOOLS & HEADLESS CHROME: THE AUTOMATION POWER-COUPLE

Yonatan Mevorach

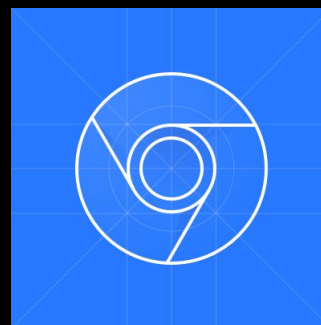


@cowchimp





WiXEngineering



“I can’t automate this”

-- no one ever

(after watching this talk)

Web Automation



Headless Browsers

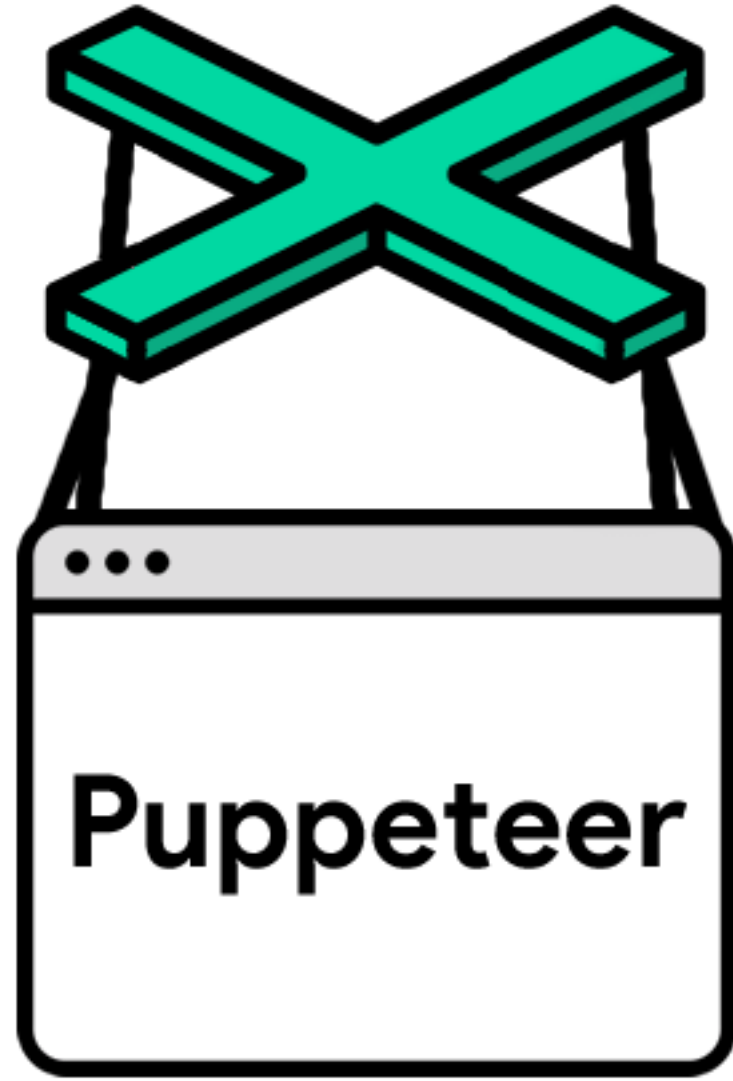




headless



Chrome





find packages



sign up or log in



★ puppeteer public

API | FAQ | Contributing

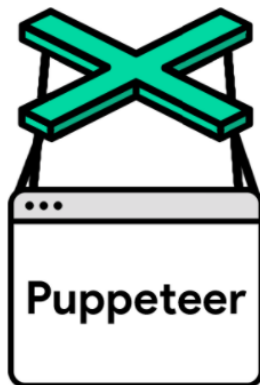
Puppeteer is a Node library which provides a high-level API to control **headless** Chrome over the **DevTools Protocol**. It can also be configured to use full (non-headless) Chrome.

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

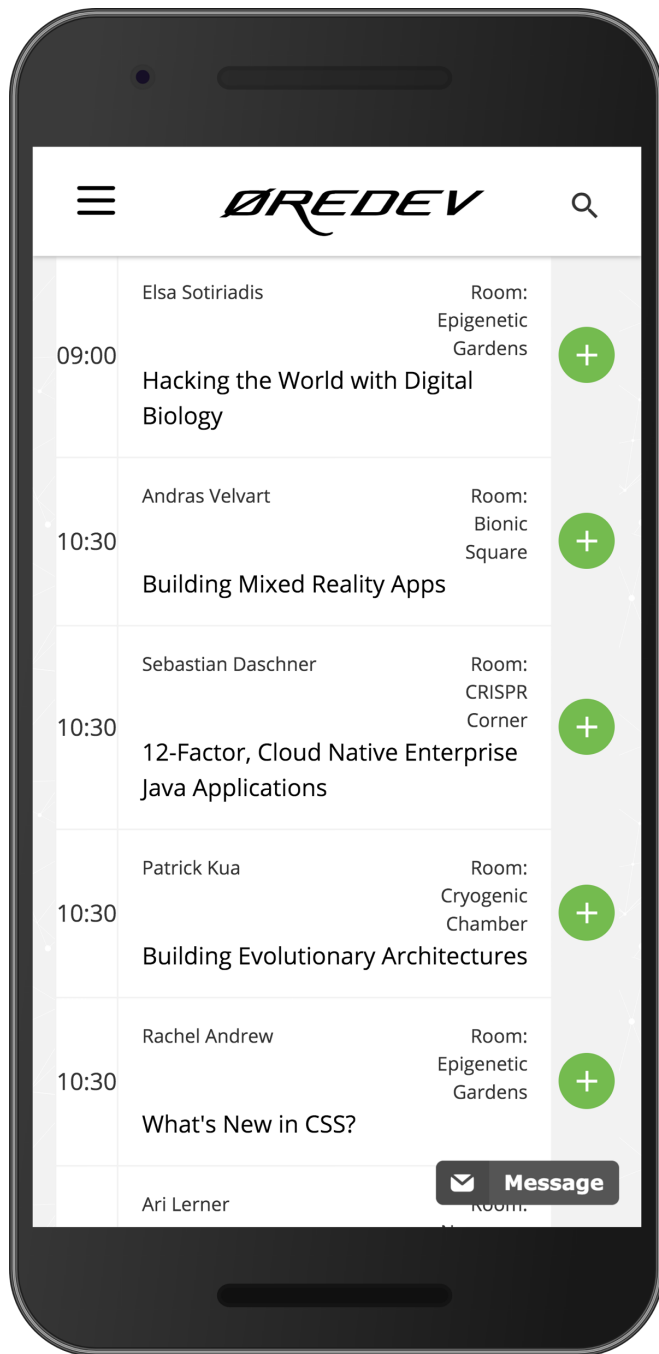
0.11.0 is the latest of 6 releases

github.com/GoogleChrome/puppeteer

Apache-2.0

Collaborators [list](#)





```
[
  { "time": "09:00", "title": "Hacking the World with Digital Biology" },
  { "time": "10:30", "title": "Building Mixed Reality Apps" },
  { "time": "10:30", "title": "12-Factor, Cloud Native Enterprise Java Applications" },
  { "time": "10:30", "title": "Building Evolutionary Architectures" },
  { "time": "10:30", "title": "What's New in CSS?" },
  { "time": "10:30", "title": "Building Performant Backends with Elixir" },
  { "time": "10:30", "title": "ASP.NET Core in Production: Lessons Learned" },
  { "time": "10:30", "title": "Stop Building Useless Software" },
  { "time": "11:30", "title": "GDPR Compliance for Your Datastore" },
  { "time": "11:30", "title": "Having Your Cake and Eating It Too – GraphQL in Reason" },
  { "time": "11:30", "title": "The New Paradigm For Leading Teams in the Digital Age" },
  { "time": "11:30", "title": "Dissecting the stdlib" },
  { "time": "11:30", "title": "Reality is Becoming Virtual" },
  { "time": "11:30", "title": "Functional Patterns for the Object Oriented" },
  { "time": "11:30", "title": "Designing Distributed Systems with TLA+" },
  { "time": "12:50", "title": "TDD on Android: Why and how?" },
  { "time": "12:50", "title": "Practice Safe Networking" },
  { "time": "12:50", "title": "Crafting Faster in the Dark" },
  { "time": "12:50", "title": "Behind the Scenes at Gapminder: A Story about Data and Visualizations" },
  { "time": "12:50", "title": "The Future of QA in the Age of AI" },
  { "time": "12:50", "title": "Blazor, a New .NET Single Page Application Framework" },
  { "time": "12:50", "title": "How to Be a High Performing Distributed Agile Team" },
  { "time": "13:50", "title": "Java with a Clojure Mindset" },
  // ...
]
```



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('http://oredev.org/2018/sessions', {
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('http://oredev.org/2018/sessions', {
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('http://oredev.org/2018/sessions', {
11.        waitUntil: 'networkidle0'
12.    });
13.
```

go to oredev schedule page

Session

oredev.org/2018/sessions

Not Secure

oredev.org/2018/sessions

☆

👤

⋮

OREDEV

TICKETS

LINE-UP

SESSIONS

FAQ

🔍

09:00

Elsa Sotiriadis
Hacking the World with Digital Biology

Room: Epigenetic Gardens

+

10:30

Andras Velvart
Building Mixed Reality Apps

Room: Bionic Square

+

✉

Leave a Message

🖱️

📄

Elements

Console

Sources

»

⋮

✕

🎬

🚫

top

▼

👁️

Filter

Default levels

▼

⚙️

>

Array.from(
 document.querySelectorAll('.session-slot')
)
.map(x => ({
 time: x.querySelector('.time').innerText,
 title: x.querySelector('.title').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('http://oredev.org/2018/sessions', {
11.    waitUntil: 'networkidle0'
12.  });
13.
14.  const schedule = await page.evaluate(() => {
15.    return Array.from(
16.      document.querySelectorAll('.session-slot')
17.    ).map(x => ({
18.      title: x.querySelector('.title').innerText
19.    }));
20.  });
21.}
```

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('http://oredev.org/2018/sessions', {
11.        waitUntil: 'networkidle0'
12.    });
13.
14.    const schedule = await page.evaluate(() => {
15.        return Array.from(
16.            document.querySelectorAll('.session-slot')
17.        ).map(x => ({
18.            time: x.querySelector('.time').innerText,
19.            title: x.querySelector('.title').innerText
20.        }));
21.    });
22.
23.    console.log(schedule);
24.}
```

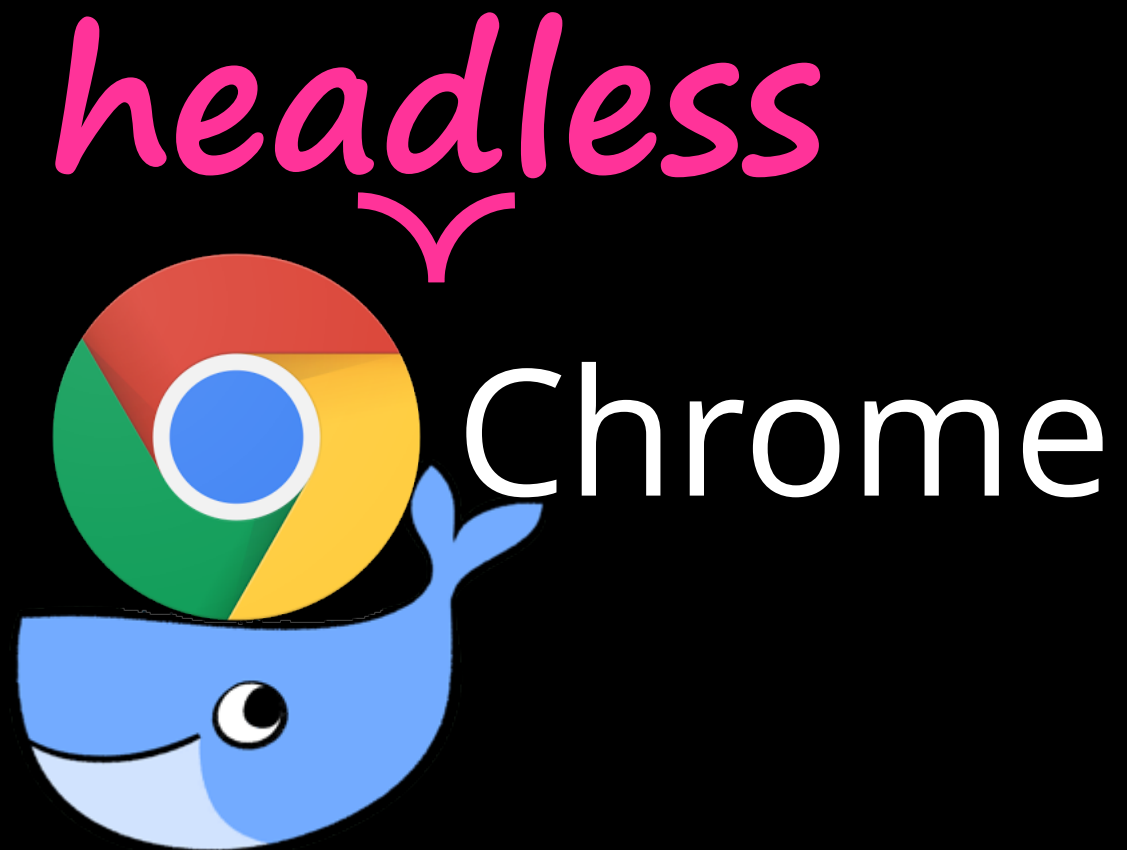
extract schedule by querying the DOM

```
10.     await page.goto('http://oredev.org/2018/sessions', {
11.         waitUntil: 'networkidle0'
12.     });
13.
14.     const schedule = await page.evaluate(() => {
15.         return Array.from(
16.             document.querySelectorAll('.session-slot')
17.         ).map(x => ({
18.             time: x.querySelector('.time').innerText,
19.             title: x.querySelector('.title').innerText
20.         }));
21.     });
22.
23.     console.log(schedule);
24.
25.     await browser.close();
```

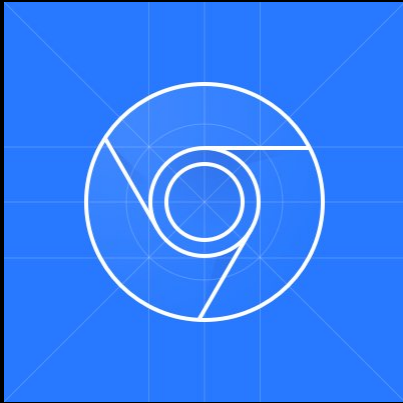
print the result

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

close the browser







Chrome DevTools

ØREDEV

TICKETS

LINE-UP

SESSIONS

FAQ

h1.message.text-center

840 × 36

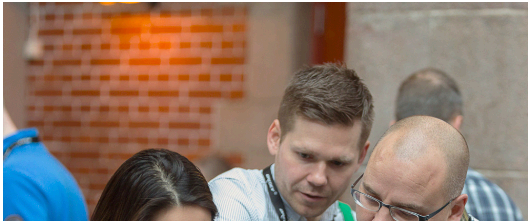
ØREDEV DEVELOPER CONFERENCE

19-22 NOV 2018

SWEDEN

GROUP DISCOUNT





Leave a Message

Elements

Console

Sources

Network

Performance

Memory

Application

Security

Audits

<div class="row">

<div class="col">

<h1 class="message text-center">...</h1> == \$0

</div>

</div>

</div>

><div class="inner-body start row row-eq-height">...</div>

><div class="inner-body row">...</div>

><div class="personas row">...</div>

><div class="inner-body">...</div>

><div class="theme">...</div>

</div>

html

body

div

div

div.inner-body.start-message

div.row

div.col

h1.message.text-center

Styles

Computed

>>

Filter

:hov .cls +

element.style {

}

.start-message

h1.message {

font-size: 30px;

}

main.css:561

h1.message {

text-transform: uppercase;

font-size: 37px;

margin: 20px 0 20px 0;

main.css:553

ØREDEV

TICKETS

LINE-UP

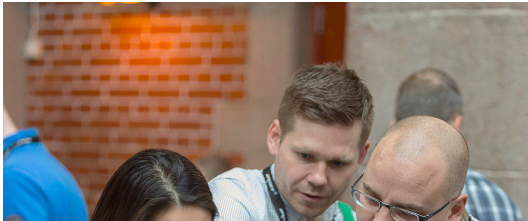
SESSIONS

FAQ

h1.message.text-center | 840 x 36

ØREDEV DEVELOPER CONFERENCE 19-22 NOV 2018 SWEDEN

GROUP DISCOUNT



Leave a Message

Elements

Console

Sources

Network

Performance

Memory

Application

Security

Audits

<div class="row">

<div class="col">

<h1 class="message text-center">...</h1> == \$0

</div>

html body div div div.inner-body.start-message div.row div.col h1.message.text-center

Styles

Computed

>>

Filter

:hov .cls +

element.style {

}

Console

Changes x

Coverage

Search

Network conditions

Rendering

Request blocking

Remote devices

*basic.css

19 19 address,dl,fieldset,p,pre,ul,ol,table {

20 margin: 0 0 20px;

20 margin: 0 0 21px;

21 21 }

22 22

23 23 ul,ol,dd {

1 insertion (+), 1 deletion (-)



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

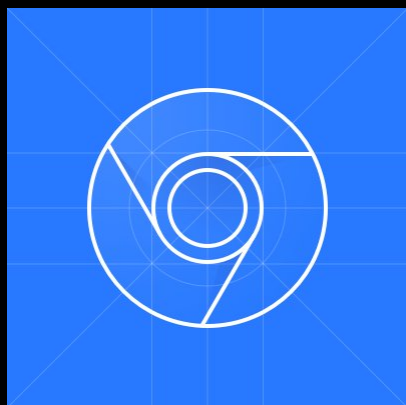
Clone or download ▼



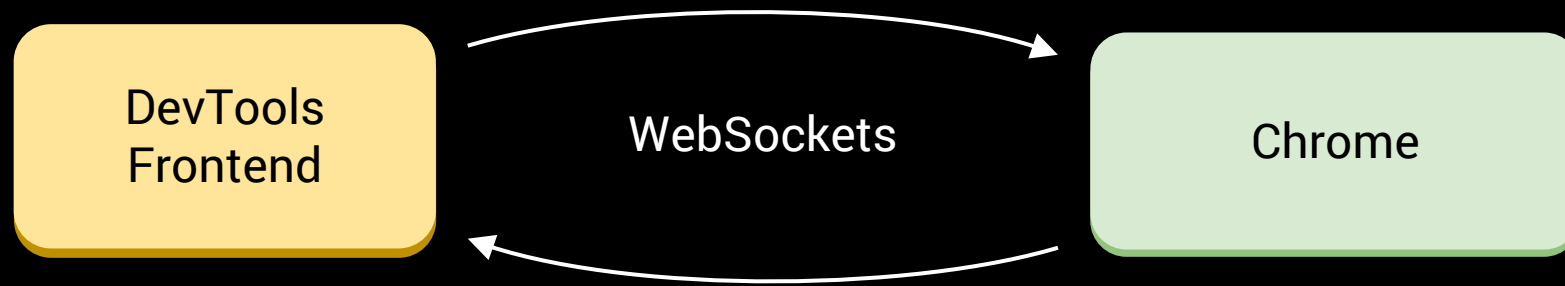
eugeneo committed with Commit Bot [DevTools] Allow the user to customize SW events ...

Latest commit 0394cbf 4 days ago

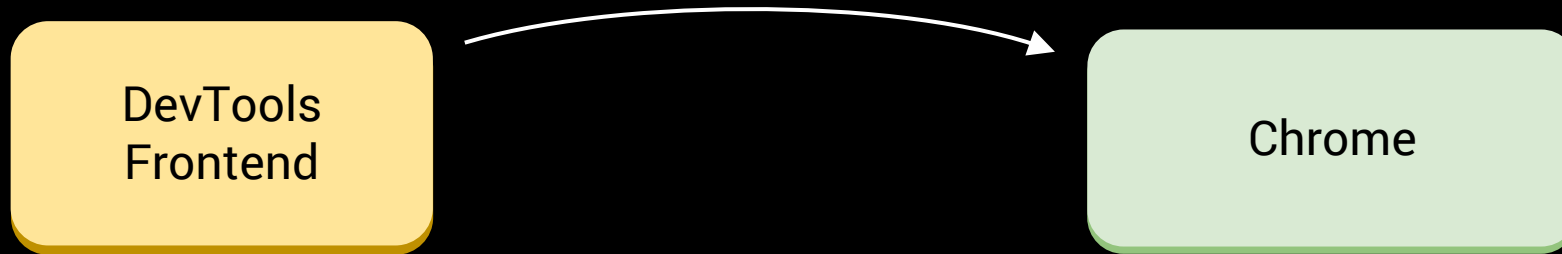
📁 front_end	[DevTools] Allow the user to customize SW events	4 days ago
📁 scripts	DevTools: move http/tests/inspector-enabled tests (part 13)	5 days ago
📁 services	[DevTools] Move main target/connection to TargetManager.	11 months ago
📄 .clang-format	DevTools: clang format all the source + clang config 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



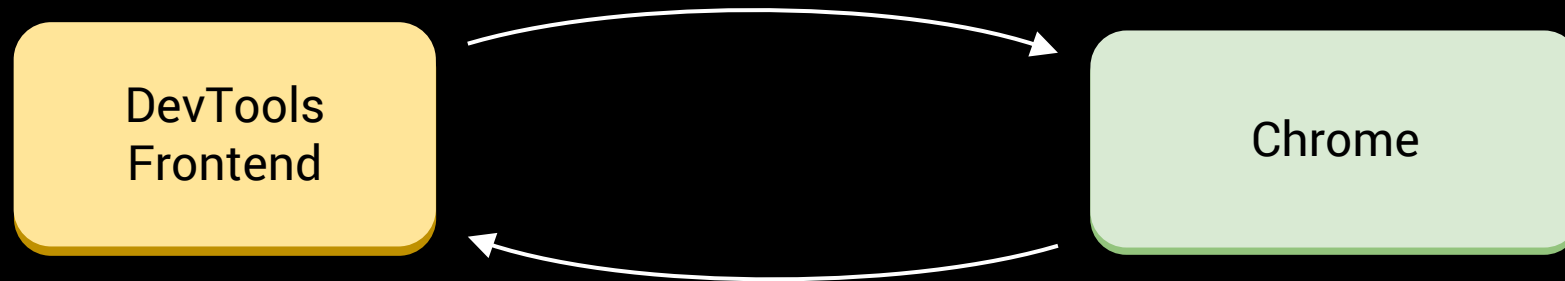
protocol
Chrome DevTools



```
{  
  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 \(1-2\)](#)

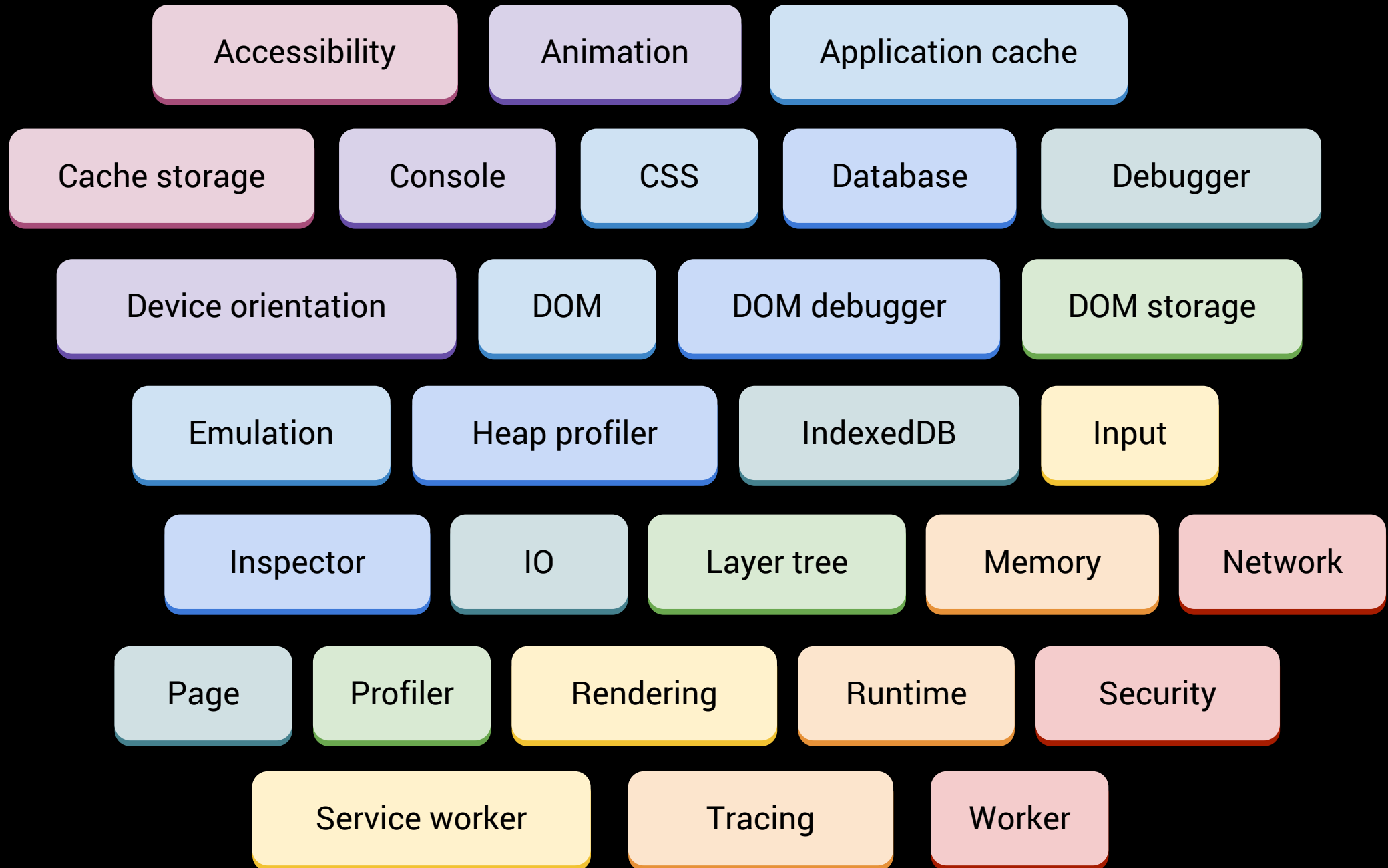
The stable release of the protocol, tagged at Chrome 54. It includes a smaller subset of the complete protocol compatibilities.

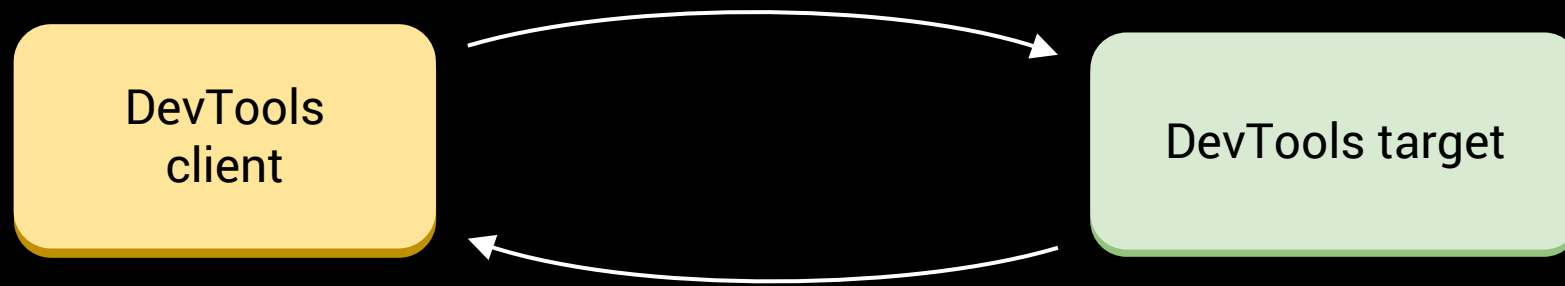
Resources

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

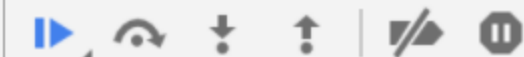





```
node --inspect index.js
```

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

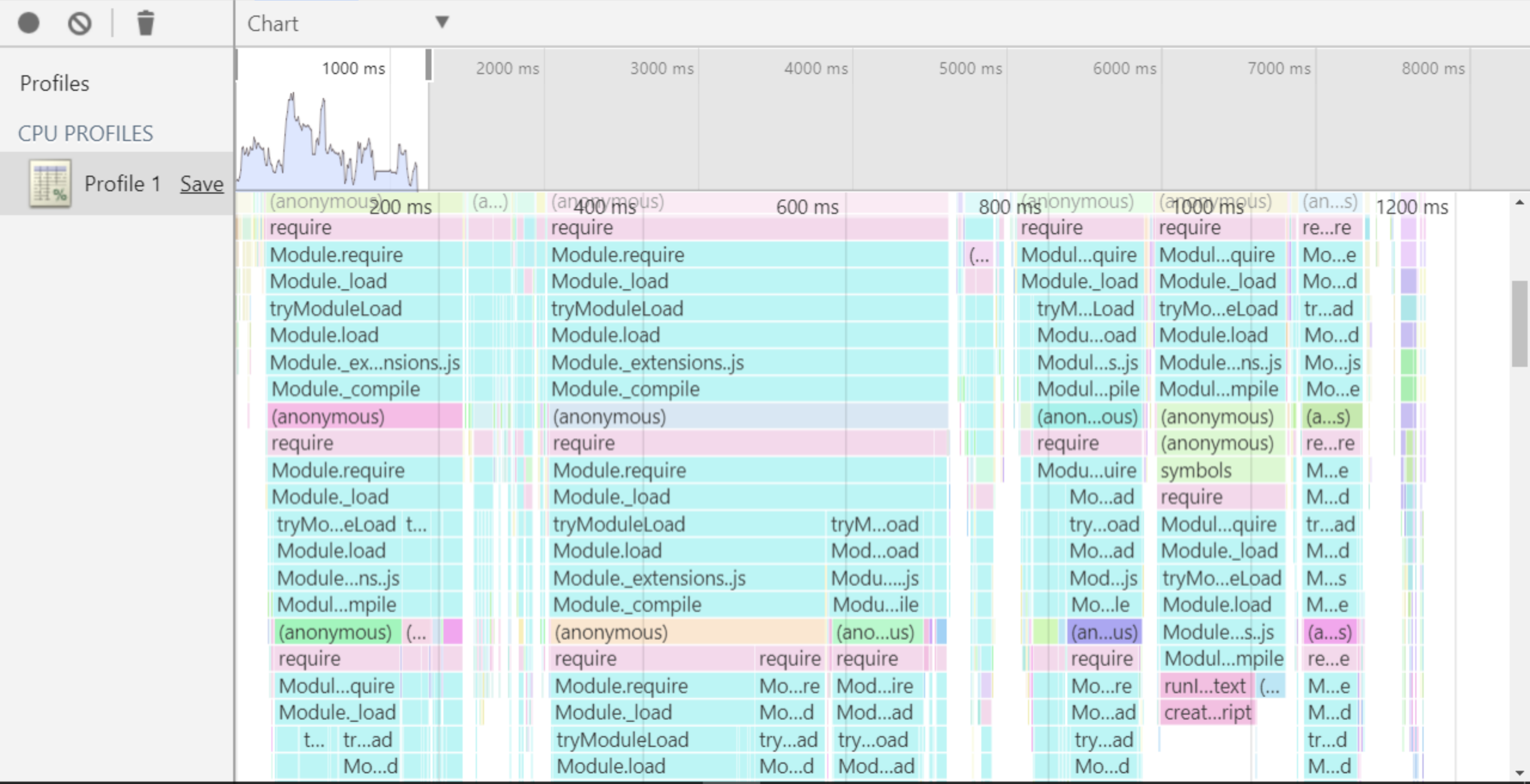
► Watch

▼ Call Stack

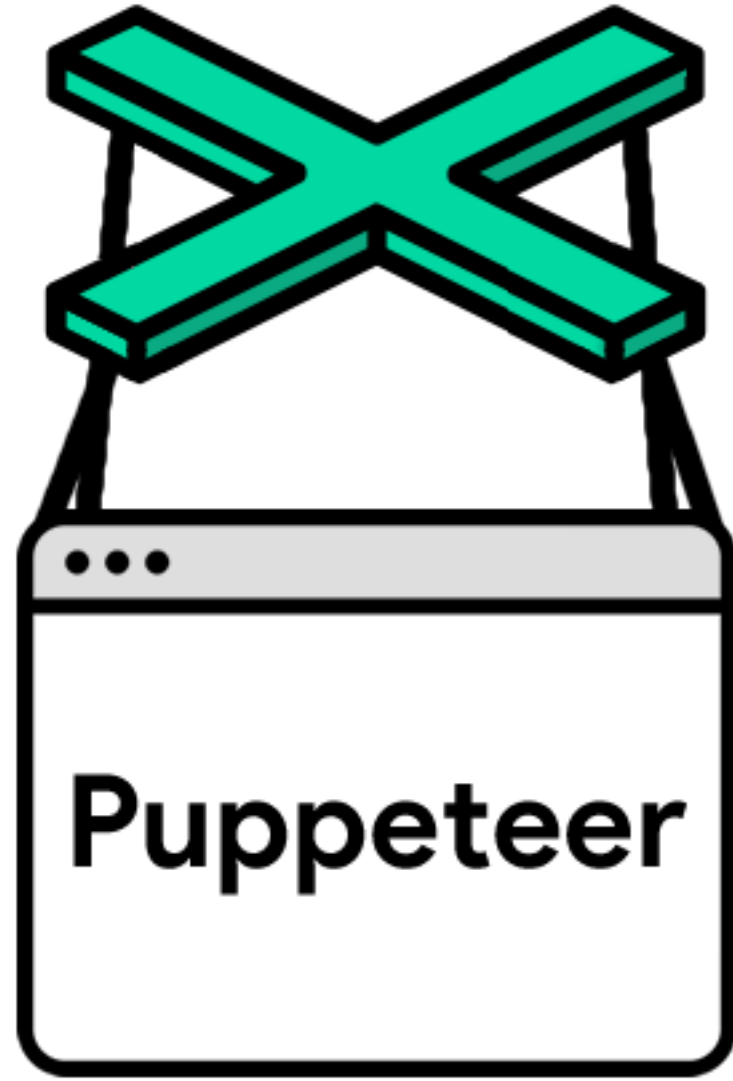
Node.js: file:///C:/_Users_Yonatan_...
paused

(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

<http://oredev.org>

chrome

--headless

--remote-debugging-port=9222

<http://oredev.org>

chrome

--headless

--remote-debugging-port=9222

<http://oredev.org>

chrome

--headless

--remote-debugging-port=9222

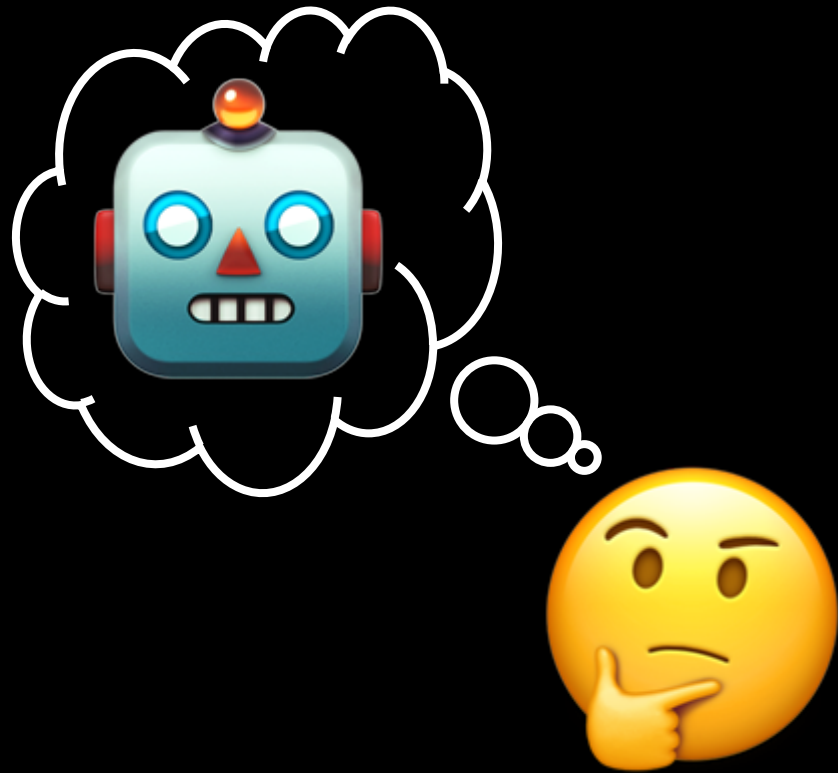
<http://oredev.org>

chrome

--headless

--remote-debugging-port=9222

<http://oredev.org>



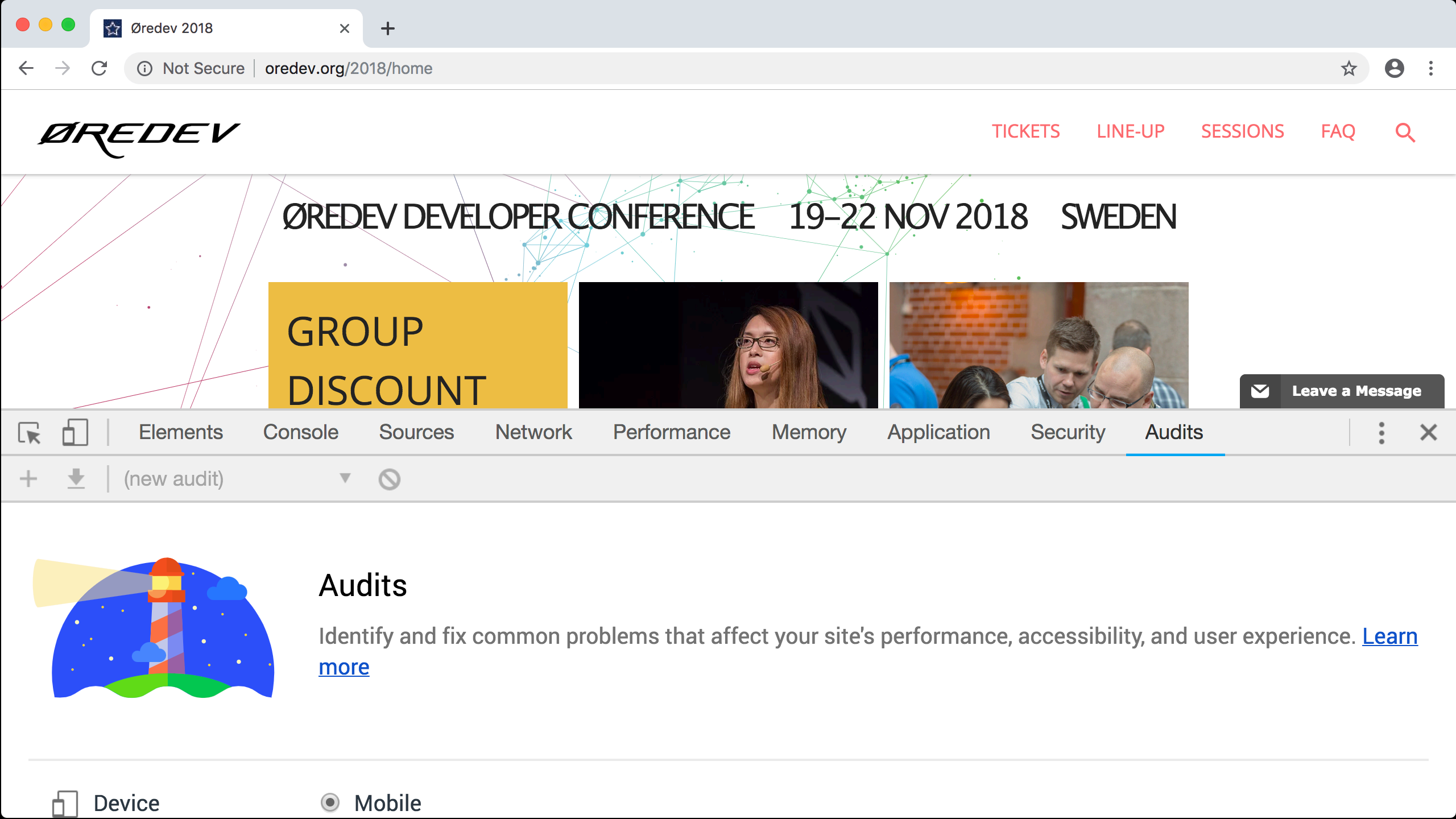
A Venn diagram illustrating the relationship between three web automation tools. It consists of three circles on a black background. The largest circle is labeled 'DevTools Protocol'. Inside it is a medium-sized circle labeled 'Selenium'. Inside the 'Selenium' circle is a smaller circle labeled 'Puppeteer'. The 'Puppeteer' circle is positioned such that it overlaps with the 'Selenium' circle, but its entire area is contained within the 'DevTools Protocol' circle.

**DevTools
Protocol**

Selenium

Puppeteer

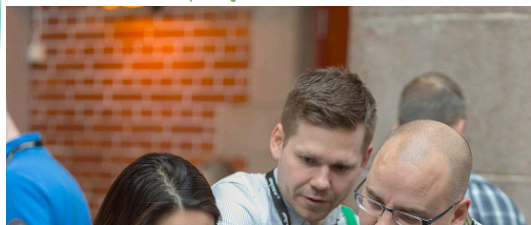
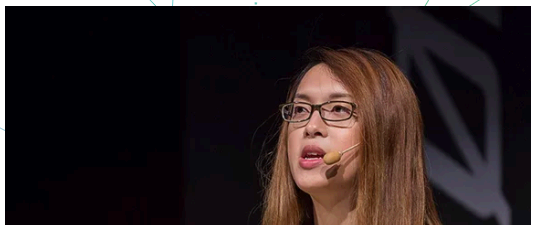




TICKETS LINE-UP SESSIONS FAQ

ØREDEV DEVELOPER CONFERENCE 19-22 NOV 2018 SWEDEN

GROUP
DISCOUNT



Leave a Message

Elements Console Sources Network Performance Memory Application Security Audits

(new audit)



Audits

Identify and fix common problems that affect your site's performance, accessibility, and user experience. [Learn more](#)

Device

Mobile

ØREDEV

TICKETS

LINE-UP

SESSIONS

FAQ

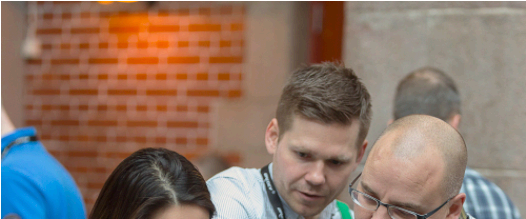
ØREDEV DEVELOPER CONFERENCE

19-22 NOV 2018

SWEDEN

GROUP DISCOUNT





Leave a Message

Elements

Console

Sources

Network

Performance

Memory

Application

Security

Audits

7:18:11 AM - oredev.org

Opportunities

These optimizations can speed up your page load.

	Opportunity		Estimated Savings
1	Serve images in next-gen formats		3.75 s
2	Efficiently encode images		3 s
3	Defer offscreen images		2.4 s





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

```
6. await c.send( 'DOM.enable' );
```

enable necessary "Domains" using

enable necessary "Domains" using raw DevTools Protocol commands

```
1. const stylesheets = [],
```

```
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-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.  
28. await browser.close();  
29.
```

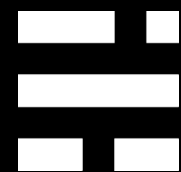
print the result and close the browser

Recap!



**tinyurl.com/
headless-talk**

 **@cowchimp**

 **blog.cowchimp.com**