

**DEVTOOLS & HEADLESS CHROME:
THE AUTOMATION POWER-COUPLE**

Yonatan Mevorach



{{company}}

 @cowchimp



“I can’t automate this”

-- no one ever

(after watching this talk)

Web Automation



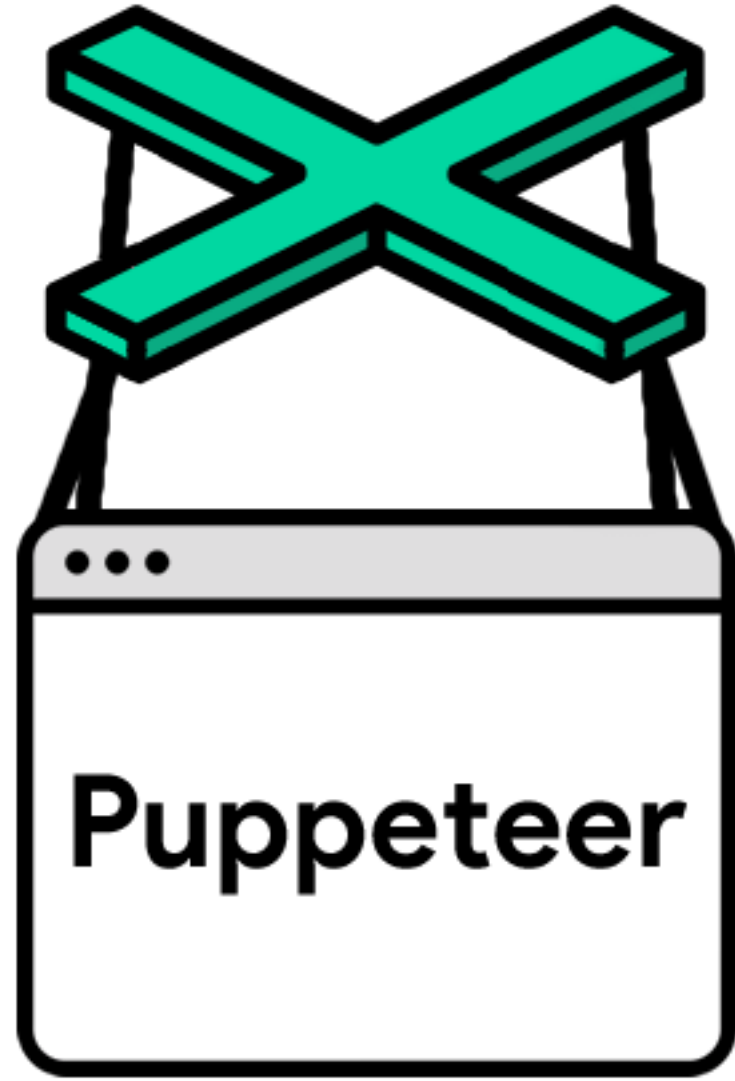
Headless Browsers





headless







find packages



sign up or log in



★ puppeteer public

API | FAQ | Contributing

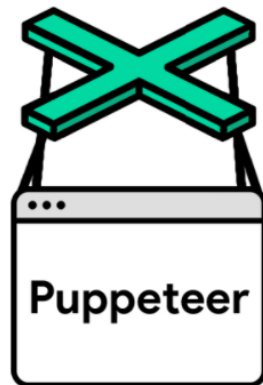
Puppeteer is a Node library which provides a high-level API to control **headless** Chrome over the **DevTools Protocol**. It can also be configured to use full (non-headless) Chrome.

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](#)



L'viv, Ukraine

Founded Nov 29, 2015

About us...

+ Invite friends

Google geeks 820

Group reviews 4

Upcoming 1

Meetups

Past Meetups 12

Our calendar 

Worldwide 695



Organizers:



Oleh Zasadnyy,

Alina Yurenko,

Diana Pinchuk, Marta

Maksymyak, Oleksandr

Pidlisnyi, Ostap Andrusiv,

The biggest Google tech conference in CEE — [GDG DevFest Ukraine 2017](#) — will take place in Lviv, on October 13-14.



Described by many as "Google I/O of Ukraine", now DevFest is the biggest Google tech conference in Central and Eastern Europe. This year, we expect over 900 engineers, managers, and entrepreneurs from over 20 countries come to DevFest to learn about Android, Web and Cloud technologies, and to experience the phenomenal atmosphere.

! Please note — for joining the event you need to get a ticket at [devfest.gdg.org.ua](#).

Why you should join us:)



Volodya Karpliuk

Co-Organizer,
Event Host



Oleksandr Pidlisnyi

Co-Organizer,
Event Host



Vitalik Zasadnyy

Co-Organizer,
Event Host



Ostap Andrusiv

Co-Organizer,
Event Host



Sofiya Huts

Co-Organizer,
Event Host



Vasylyna Mytsak

Co-Organizer,
Event Host



Oleh Zasadnyy

Organizer,
Event Host



Diana Pinchuk

Co-Organizer,

Log in

Not registered with us yet? [Sign up](#)

Email address:

Password:

[Forgot your password?](#)

Keep me signed in

[Create a Meetup](#)[Log in](#)[Help](#) [About Us](#) [Meetup Pro](#) [Jobs](#) [Apps](#) [API](#) [Topics](#) [Browse Cities](#) [Blog](#)

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 logInToMeetup(page);
11.
12.  const u = 'https://www.meetup.com/GDG-Lviv/events/240383203/';
13.  await page.goto(u);
```


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

2.

3. `init();`

4.

`require Puppeteer`

```
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

```
12.   const u = 'https://www.meetup.com/GDG-LVIV/events/240383203/';
```

```
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 logInToMeetup(page);
11.
```

open a new tab

```
14.
```

```
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 logInToMeetup(page);
11.
12.  const u = 'https://www.meetup.com/GDG-Lviv/events/240383203/';
13.  await page.goto(u);
```

log in to get data visible to members only

```
16.   return Array.from(
```

```
15.     const names = await page.evaluate(function() {
16.         return Array.from(
17.             document.querySelectorAll('#rsvp-list .member-name')
18.         ).map(x => x.innerText);
19.     });
20.
21.     console.log(names);
22.
23.     await browser.close();
24. }
25.
26. async function logInToMeetup(page) {
27.     await page.goto('https://secure.meetup.com/login/');
28.
29.     await page.focus('#email');
30.     await page.type('meetup.user@gmail.com');
```

go to log in page

```
33.
```



```
18.     ).map(x => x.innerText);
19.   });
20.
21.   console.log(names);
22.
23.   await browser.close();
24. }
25.
26. async function logInToMeetup(page) {
27.   await page.goto('https://secure.meetup.com/login/');
28.
29.   await page.focus('#email');
30.   await page.type('meetup.user@gmail.com');
31.   await page.focus('#password');
32.   await page.type('p@$w0rd');
33.
```

fill in the user email

```
36.   await page.waitForNavigation().
```

```
20.
21.     console.log(names);
22.
23.     await browser.close();
24. }
25.
26. async function logIntoMeetup(page) {
27.     await page.goto('https://secure.meetup.com/login/');
28.
29.     await page.focus('#email');
30.     await page.type('meetup.user@gmail.com');
31.     await page.focus('#password');
32.     await page.type('p$$w0rd');
33.
34.     await page.click('[name="submitButton"]');
35.
```

fill in the user password

```
22.  
23.     await browser.close();  
24. }  
25.  
26. async function logIntoMeetup(page) {  
27.     await page.goto('https://secure.meetup.com/login/');  
28.  
29.     await page.focus('#email');  
30.     await page.type('meetup.user@gmail.com');  
31.     await page.focus('#password');  
32.     await page.type('p@$w0rd');  
33.  
34.     await page.click('[name="submitButton"]');  
35.  
36.     await page.waitForNavigation();  
37. }
```

click the "Log in" button

```
23.     await browser.close();
24. }
25.
26. async function logInToMeetup(page) {
27.     await page.goto('https://secure.meetup.com/login/');
28.
29.     await page.focus('#email');
30.     await page.type('meetup.user@gmail.com');
31.     await page.focus('#password');
32.     await page.type('p@$w0rd');
33.
34.     await page.click('[name="submitButton"]');
35.
36.     await page.waitForNavigation();
37. }
```

wait for a redirect to happen

```
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 logIntoMeetup(page);
11.
12.  const u = 'https://www.meetup.com/GDG-Lviv/events/240383203/';
13.  await page.goto(u);
14.
15.  const names = await page.evaluate(function() {
16.    return Array.from(
```

go to specific meetup page

```
19.   });
```



```
3.  init();
4.
5.  async function init() {
6.    const browser = await puppeteer.launch();
7.
8.    const page = await browser.newPage();
9.
10.   await logIntoMeetup(page);
11.
12.   const u = 'https://www.meetup.com/GDG-Lviv/events/240383203/';
13.   await page.goto(u);
14.
15.   const names = await page.evaluate(function() {
16.     return Array.from(
17.       document.querySelectorAll('#rsvp-list .member-name')
18.     ).map(x => x.innerText);
```

execute JS on the client

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

```
5. async function init() {
6.   const browser = await puppeteer.launch();
7.
8.   const page = await browser.newPage();
9.
10.  await logIntoMeetup(page);
11.
12.  const u = 'https://www.meetup.com/GDG-Lviv/events/240383203/';
13.  await page.goto(u);
14.
15.  const names = await page.evaluate(function() {
16.    return Array.from(
17.      document.querySelectorAll('#rsvp-list .member-name')
18.    ).map(x => x.innerText);
19.  });
20.
```

fetch attendee names with DOM traversal

```
23.   await browser.close();
```

```
9.
10.  await logIntoMeetup(page);
11.
12.  const u = 'https://www.meetup.com/GDG-Lviv/events/240383203/';
13.  await page.goto(u);
14.
15.  const names = await page.evaluate(function() {
16.    return Array.from(
17.      document.querySelectorAll('#rsvp-list .member-name')
18.    ).map(x => x.innerText);
19.  });
20.
21.  console.log(names);
22.
23.  await browser.close();
24. }
```

print the result

```
27.  await page.goto('https://secure.meetup.com/login/');
```

```
11.
12.   const u = 'https://www.meetup.com/GDG-Lviv/events/240383203/';
13.   await page.goto(u);
14.
15.   const names = await page.evaluate(function() {
16.     return Array.from(
17.       document.querySelectorAll('#rsvp-list .member-name')
18.     ).map(x => x.innerText);
19.   });
20.
21.   console.log(names);
22.
23.   await browser.close();
24. }
25.
26. async function logInToMeetup(page) {
```

close the browser

```
29.   await page.focus( '#email' );
```

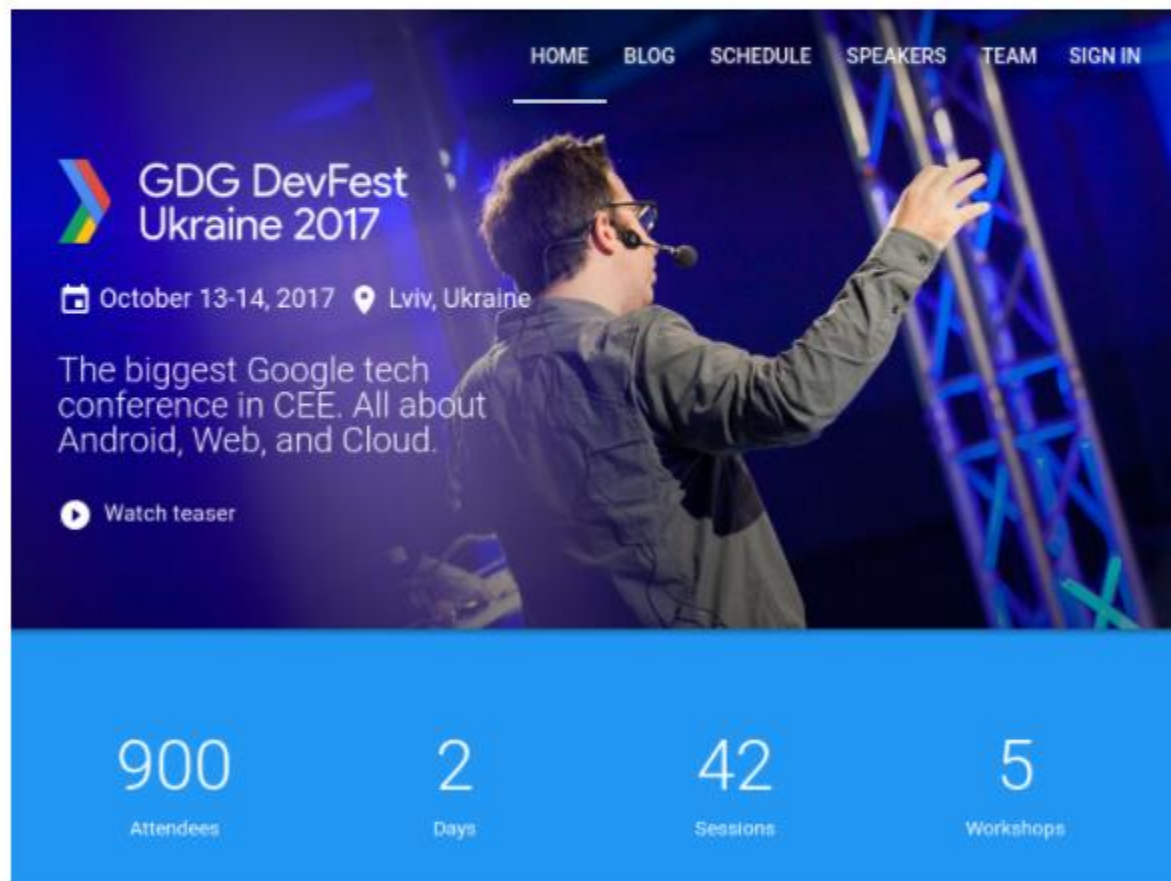



```
1 const browser = await puppeteer.launch();
2 const page = await browser.newPage();
3 await page.goto('https://devfest.gdg.org.ua/');
4 await page.screenshot({path: 'example.png'});
5 await browser.close();
```

 RUN IT

Note: There's no need to `require('puppeteer')` in the editor. It's done for you!

RESULT



headless



Chrome

headless



Firefox



Chrome DevTools

div#sizedImgDiv | 240 x 80



October 13-14, 2017 Lviv, Ukraine

The biggest Google tech conference in CEE. All about Android, Web, and Cloud.

Watch teaser



Elements Console Network Performance Sources Memory Application Security Audits

```

<iron-image class="logo" src="/images/logo-light.svg" sizing="contain" alt="GDG DevFest Ukraine 2017">
  #shadow-root (open)
    <style scope="iron-image-0">...</style>
    <div id="sizedImgDiv" role="img" aria-label="GDG DevFest Ukraine 2017" style="background-image: url('/images/logo-light.svg'); background-size: contain; background-position: center center; background-repeat: no-repeat;">
    </div> == $0
    
    <div id="placeholder" class hidden style="background-size: contain; background-position: center center; background-
  
```

Styles Computed Event Listeners >>

Filter :hov .cls +

```

element.style {
  background-image: url(/images/logo-light.svg);
  background-size: contain;
  background-position: center center;
  background-repeat: no-repeat;
}
:host([sizing]) <style>...</style>
#sizedImgDiv {
  display: block;

```

... animatable-content div div iron-image #shadow-root div#sizedImgDiv


```

    <iron-image class="logo" src="/images/logo-light.svg" sizing=
    "contain" alt="GDG DevFest Ukraine 2017">
      #shadow-root (open)
        <style scope="iron-image-0">...</style>
        <div id="sizedImgDiv" role="img" aria-label="GDG DevFest
        Ukraine 2017" style="background-image: url("/images/logo-
        light.svg"); background-size: contain; background-
        position: center center; background-repeat: no-repeat;">
        </div> == $0
  
```

Styles Computed Event Listeners >>

Filter :hov .cls +

```

element.style {
  background-image:
    url(/images/logo-light.svg);
  background-size: contain;
  background-position: center
  center;
  background-repeat: no-repeat;
}
  
```

... animatable-content div div iron-image #shadow-root **div#sizedImgDiv**

URL filter Content scripts

URL	Type	Total Bytes	Unused Bytes	
https://devfest.gdg.org.ua/src/hoverboard-app.html	JS	755 789	374 984 49.6 %	
htt... /modules-3474bcdcf0768a446d3f92f4f03f2dd78.js	JS	347 943	133 782 38.4 %	
https://maps.googleapis.com/maps-api-v3/api.../util.js	JS	133 926	88 721 66.2 %	
https://maps.googleapis.com/maps-api-v.../controls.js	JS	78 542	42 164 53.7 %	
https://maps.googleapis.com/maps-api-... /common.js	JS	116 775	38 804 33.2 %	

823 KB of 1.8 MB bytes are not used. (45%)



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

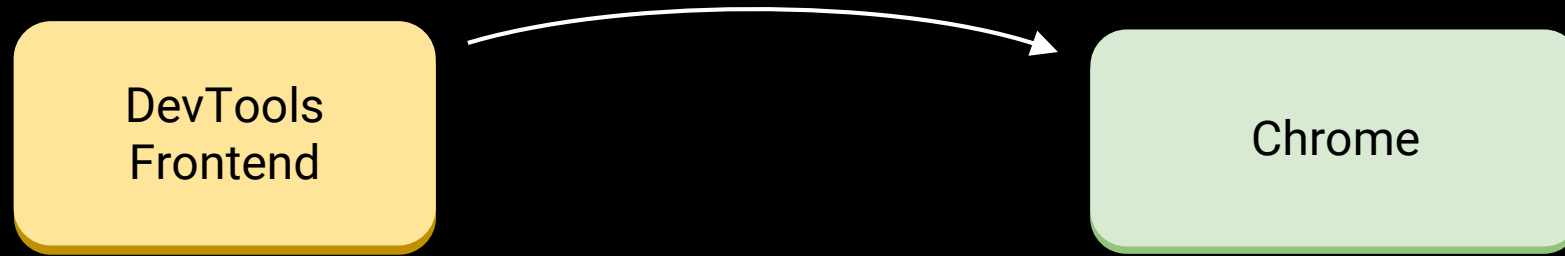


Chrome DevTools

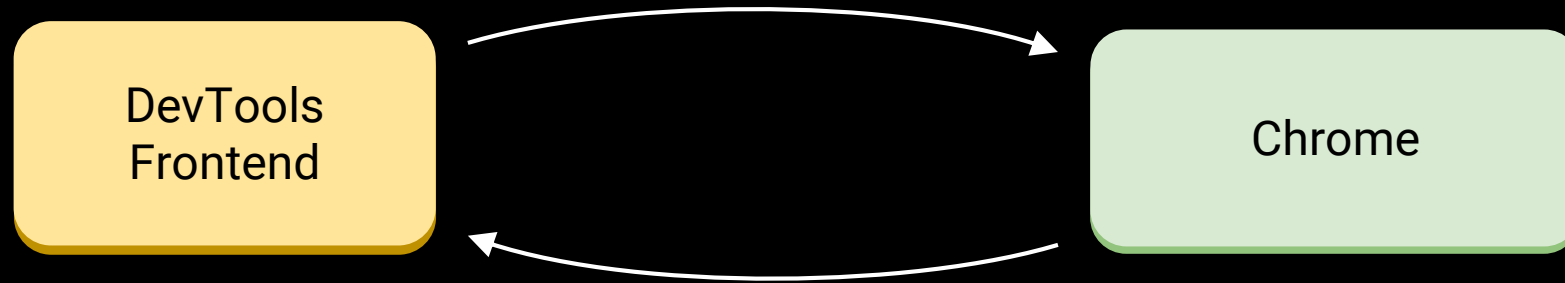
protocol



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

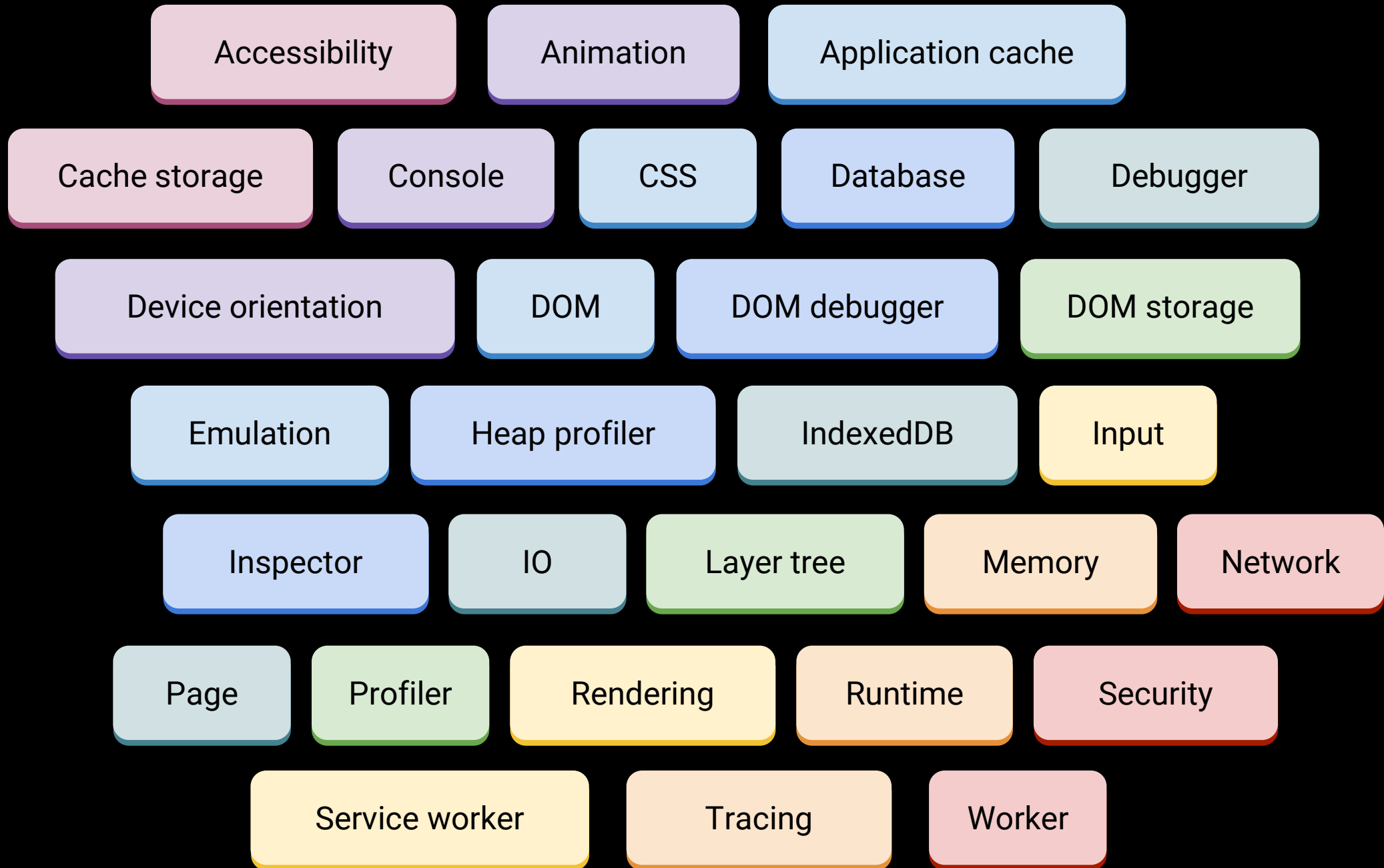
The stable release of the protocol, tagged at Chrome 54. It includes a smaller subset of the complete protocol compatibilities.

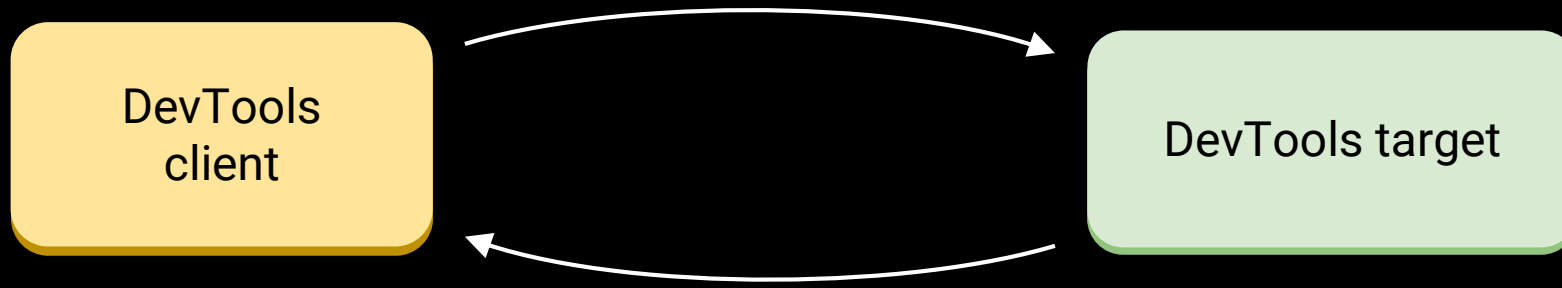
Resources

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

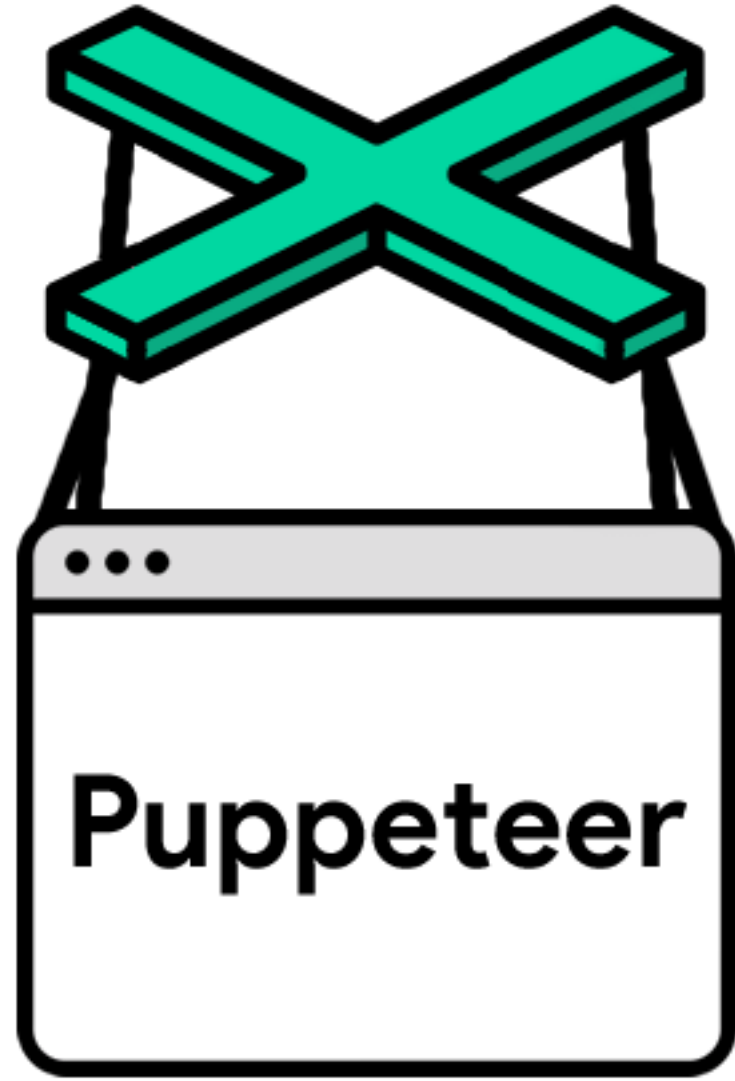
Watch

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

--disable-gpu

--remote-debugging-port=9222

<https://devfest.gdg.org.ua>

chrome

--headless

--disable-gpu

--remote-debugging-port=9222

<https://devfest.gdg.org.ua>

chrome

--headless

--disable-gpu

--remote-debugging-port=9222

<https://devfest.gdg.org.ua>

chrome

--headless

--disable-gpu

--remote-debugging-port=9222

<https://devfest.gdg.org.ua>

chrome

--headless

--disable-gpu

--remote-debugging-port=9222

<https://devfest.gdg.org.ua>

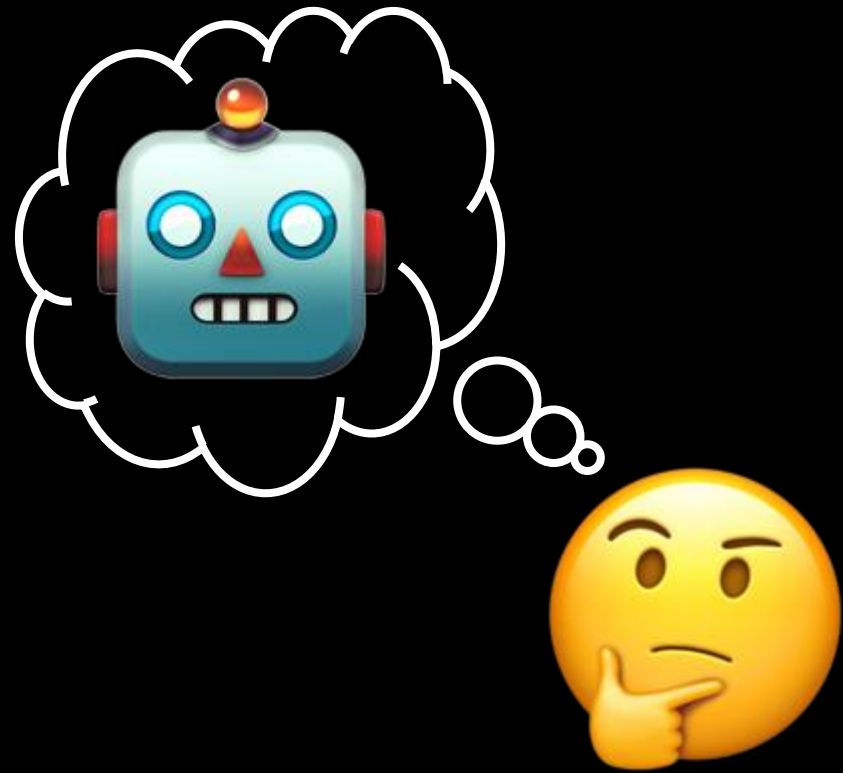
chrome

--headless

--disable-gpu

--remote-debugging-port=9222

<https://devfest.gdg.org.ua>



A Venn diagram illustrating the relationship between three web automation tools. It consists of three overlapping circles on a black background. The largest circle, at the top, is labeled 'DevTools Protocol'. A medium-sized circle, labeled 'Selenium', overlaps with the bottom of the 'DevTools Protocol' circle. A smaller circle, labeled 'Puppeteer', overlaps with the bottom of the 'Selenium' circle. The labels are in white, bold, sans-serif font.

**DevTools
Protocol**

Selenium

Puppeteer





October 13-14, 2017 Lviv, Ukraine



Elements Console Network Performance Sources Memory Application Security Audits



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

Perform an audit...



October 13-14, 2017 Lviv, Ukraine



Best Practices

77

We've compiled some recommendations for modernizing your web app and avoiding performance pitfalls. These audits do not affect your score but are worth a look.

3 failed audits

- ▶ Does not use passive listeners to improve scrolling performance ✗
- ▶ Does not open external anchors using rel="noopener" ✗
- ▶ Uses deprecated API's: 2 warnings found ✗

▶ 10 Passed Audits

- ▶ Avoids Application Cache ✓
- ▶ Avoids WebSQL DB ✓
- ▶ Uses HTTPS ✓





Unused CSS Calculator

```
1. const puppeteer = require('puppeteer');
2.
3. init();
4.
5. async function init() {
6.   const browser = await puppeteer.launch();
7.
8.   const unusedCSS = await fetchUnusedCssPercentage(browser);
9.
10.  console.log(`${unusedCSS}% of your CSS is unused`);
11.
12.  await browser.close();
```



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

launch Chrome with Puppeteer

```
1. const puppeteer = require('puppeteer');
2.
3. init();
4.
5. async function init() {
6.   const browser = await puppeteer.launch();
7.
8.   const unusedCSS = await fetchUnusedCssPercentage(browser);
9.
10.  console.log(`${unusedCSS}% of your CSS is unused`);

```

fetch the percentage of unused CSS

```
3.  init();
4.
5.  async function init() {
6.    const browser = await puppeteer.launch();
7.
8.    const unusedCSS = await fetchUnusedCssPercentage(browser);
9.
10.   console.log(` ${unusedCSS}% of your CSS is unused`);
11.
12.   await browser.close();
13. }
14.
15. async function fetchUnusedCssPercentage(browser) {
16.   const page = await browser.newPage();
17.
18.   const cp = page._client;
19.   await cp.send('Page.enable');
20.   await cp.send('DOM.enable');
```



```
4.
5. async function init() {
6.   const browser = await puppeteer.launch();
7.
8.   const unusedCSS = await fetchUnusedCssPercentage(browser);
9.
10.  console.log(`${unusedCSS}% of your CSS is unused`);
11.
12.  await browser.close();
13. }
14.
15. async function fetchUnusedCssPercentage(browser) {
16.   const page = await browser.newPage();
17.
18.   const cn = page.client;
```

open a new tab

```
21.   await cn.send('CSS enable');
```

```
8.   const unusedCSS = await fetchUnusedCssPercentage(browser);
9.
10.  console.log(` ${unusedCSS}% of your CSS is unused`);
11.
12.  await browser.close();
13. }
14.
15. async function fetchUnusedCssPercentage(browser) {
16.   const page = await browser.newPage();
17.
18.   const cp = page._client;
19.   await cp.send('Page.enable');
20.   await cp.send('DOM.enable');
21.   await cp.send('CSS.enable');
```

enable necessary "Domains" using raw DevTools Protocol commands

24.

```
11.
12.   await browser.close();
13. }
14.
15. async function fetchUnusedCssPercentage(browser) {
16.   const page = await browser.newPage();
17.
18.   const cp = page._client;
19.   await cp.send('Page.enable');
20.   await cp.send('DOM.enable');
21.   await cp.send('CSS.enable');
22.
23.   await cp.send('CSS.startRuleUsageTracking');
24.
25.   const stylesheets = [];
```

start tracking CSS usage

```
15. async function fetchUnusedCssPercentage(browser) {
16.     const page = await browser.newPage();
17.
18.     const cp = page._client;
19.     await cp.send('Page.enable');
20.     await cp.send('DOM.enable');
21.     await cp.send('CSS.enable');
22.
23.     await cp.send('CSS.startRuleUsageTracking');
24.
25.     const stylesheets = [];
26.     cp.on('CSS.styleSheetAdded', stylesheet => {
27.         stylesheets.push(stylesheet.header);
28.     });
```

listen for when a stylesheet is loaded

```
31.     await page.goto(url);
```

```
21.   await cp.send('CSS.enable');
22.
23.   await cp.send('CSS.startRuleUsageTracking');
24.
25.   const stylesheets = [];
26.   cp.on('CSS.styleSheetAdded', stylesheet => {
27.     stylesheets.push(stylesheet.header);
28.   });
29.
30.   const url = 'https://example-site-izvafmtnha.now.sh/';
31.   await page.goto(url);
32.   await page.click('.tab.type2');
33.   await page.click('.tab.type3');
34.   await page.click('.tab.type4');
```

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

```
37.
```

```
24.  
25.  const stylesheets = [];  
26.  cp.on('CSS.styleSheetAdded', stylesheet => {  
27.    stylesheets.push(stylesheet.header);  
28.  });  
29.  
30.  const url = 'https://example-site-izvafmtnha.now.sh/';  
31.  await page.goto(url);  
32.  await page.click('.tab.type2');  
33.  await page.click('.tab.type3');  
34.  await page.click('.tab.type4');  
35.  
36.  const {ruleUsage} = await cp.send('CSS.stopRuleUsageTracking');  
37.  
38.  return calcUsedCssPercentage(stylesheets, ruleUsage);
```

stop tracking CSS usage and fetch used rules

```
41.  function calcUsedCssPercentage(stylesheets, ruleUsage) {
```

```
26. cp.on('CSS.stylesheetAdded', stylesheet => {
27.   stylesheets.push(stylesheet.header);
28. });
29.
30. const url = 'https://example-site-izvafmtinha.now.sh/';
31. await page.goto(url);
32. await page.click('.tab.type2');
33. await page.click('.tab.type3');
34. await page.click('.tab.type4');
35.
36. const {ruleUsage} = await cp.send('CSS.stopRuleUsageTracking');
37.
38. return calcUnusedCssPercentage(stylesheets, ruleUsage);
39. }
40.
```

crunch the numbers

```
43. stylesheets.forEach(function(stylesheet) {
```

```
1. const puppeteer = require('puppeteer');
2.
3. init();
4.
5. async function init() {
6.   const browser = await puppeteer.launch();
7.
8.   const unusedCSS = await fetchUnusedCssPercentage(browser);
9.
10.  console.log(` ${unusedCSS}% of your CSS is unused` );
11.
12.  await browser.close();
13. }
```

print the result and close the browser

```
16.   const page = await browser.newPage();
```


Make it generic

```
1. const puppeteer = require('puppeteer');
2.
3. init();
4.
5. async function init() {
6.   const browser = await puppeteer.launch();
7.
8.   const unusedCSS = await fetchUnusedCssPercentage(
9.     browser,
10.    async function(page) {
11.      const url = 'https://example-site-izvafmtnha.now.sh/';
12.      await page.goto(url);
13.      await page.click('.tab.type2');
14.      await page.click('.tab.type3');
15.      await page.click('.tab.type4');
16.    }
```

pass case-specific interaction instructions as callback

```
19.   console.log(`${unusedCSS}% of your CSS is unused`);
```

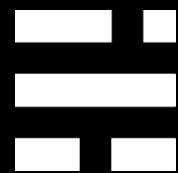
Recap!



**tinyurl.com/
headless-talk**



@cowchimp



blog.cowchimp.com