HEISENBUG

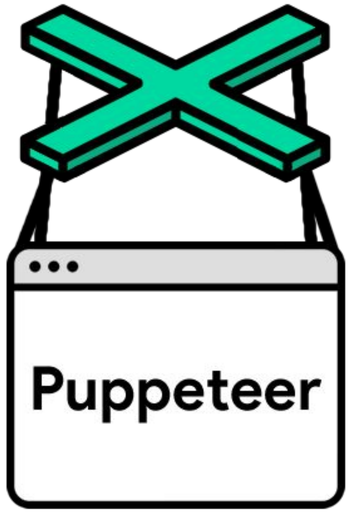
Heisenbug 2019

Modern Web Testing And Automation with Puppeteer

Agenda

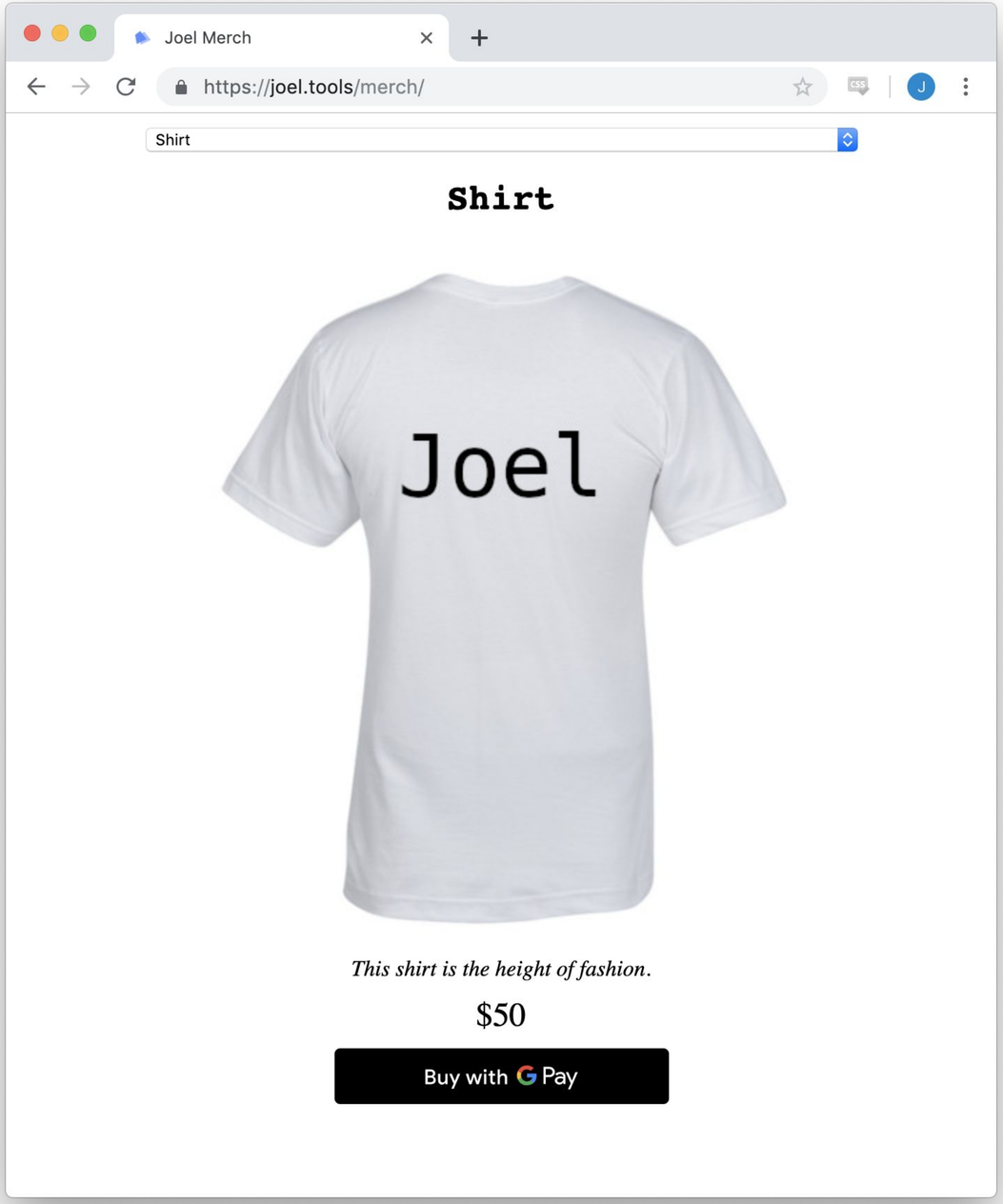
- 👉 What is Puppeteer?
- 👉 Announcement 🎁
- 👉 Testing Fundamentals
- 👉 Testing Modern Web
- 👉 Community 👫

HEISENBUG



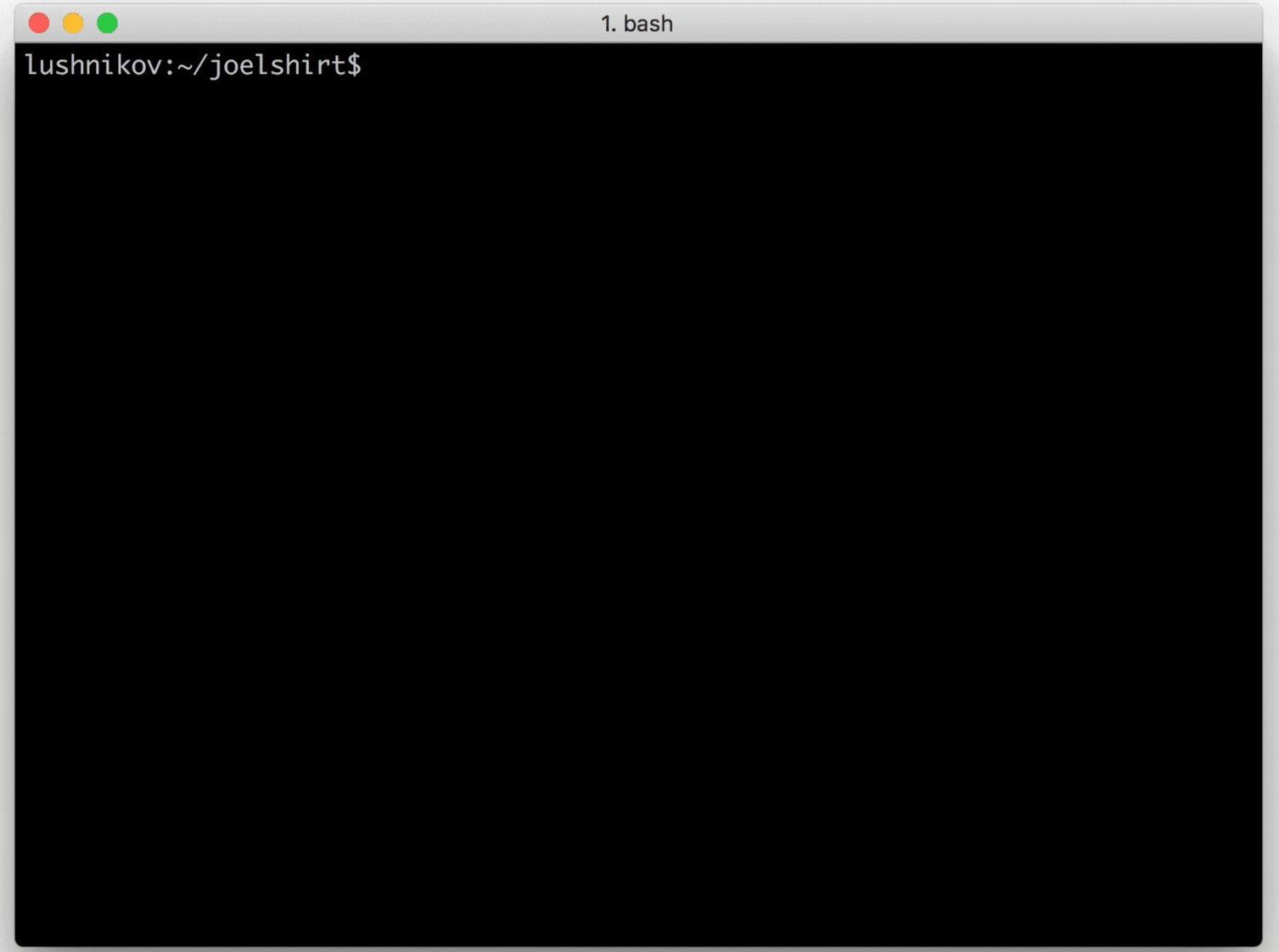
What is Puppeteer?

- 👉 Browser Automation Library
- 👉 Puppeteer = Node.js + Chrome
- 👉 Open pages, navigate to websites, evaluate javascript

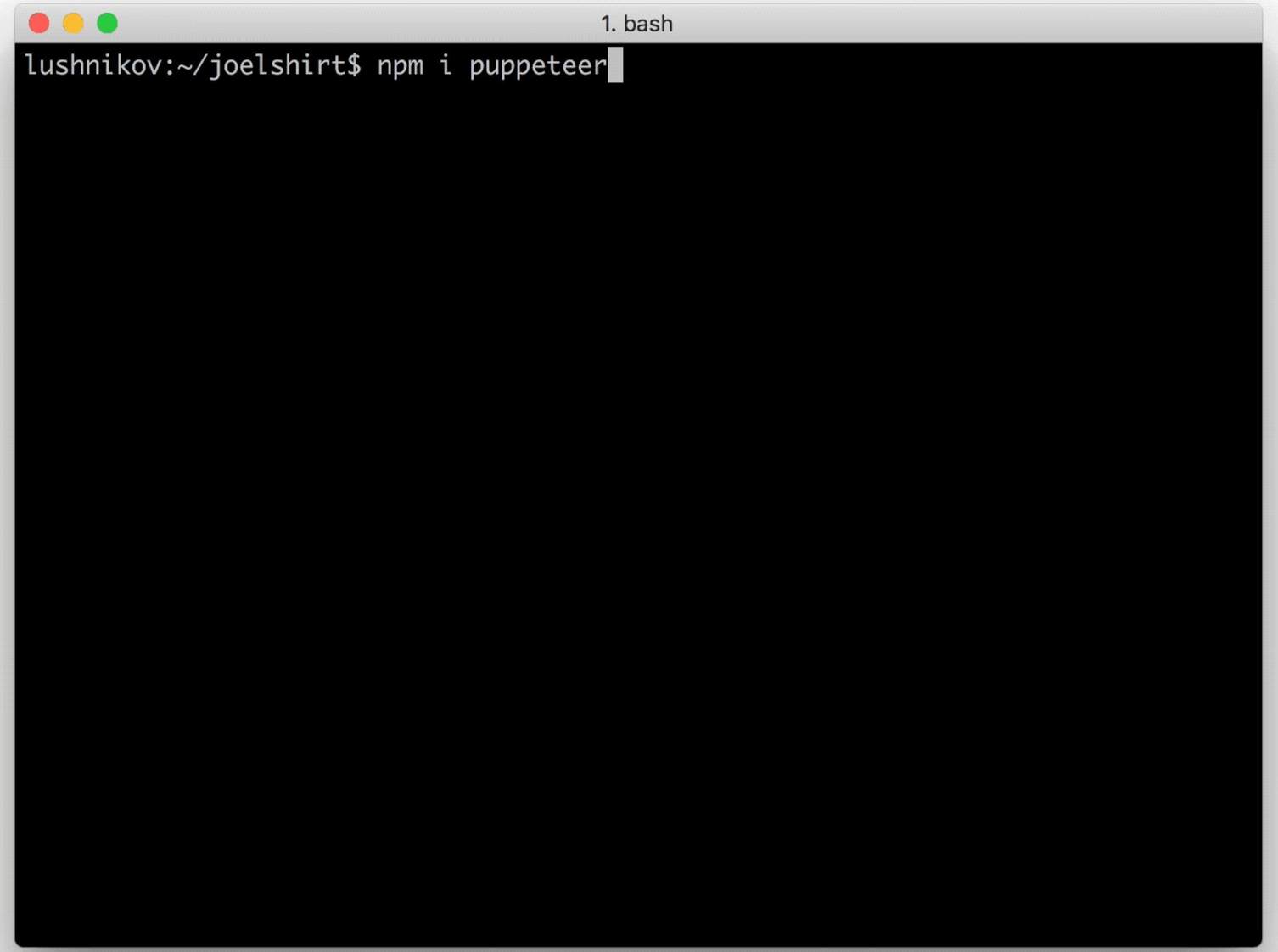


$50

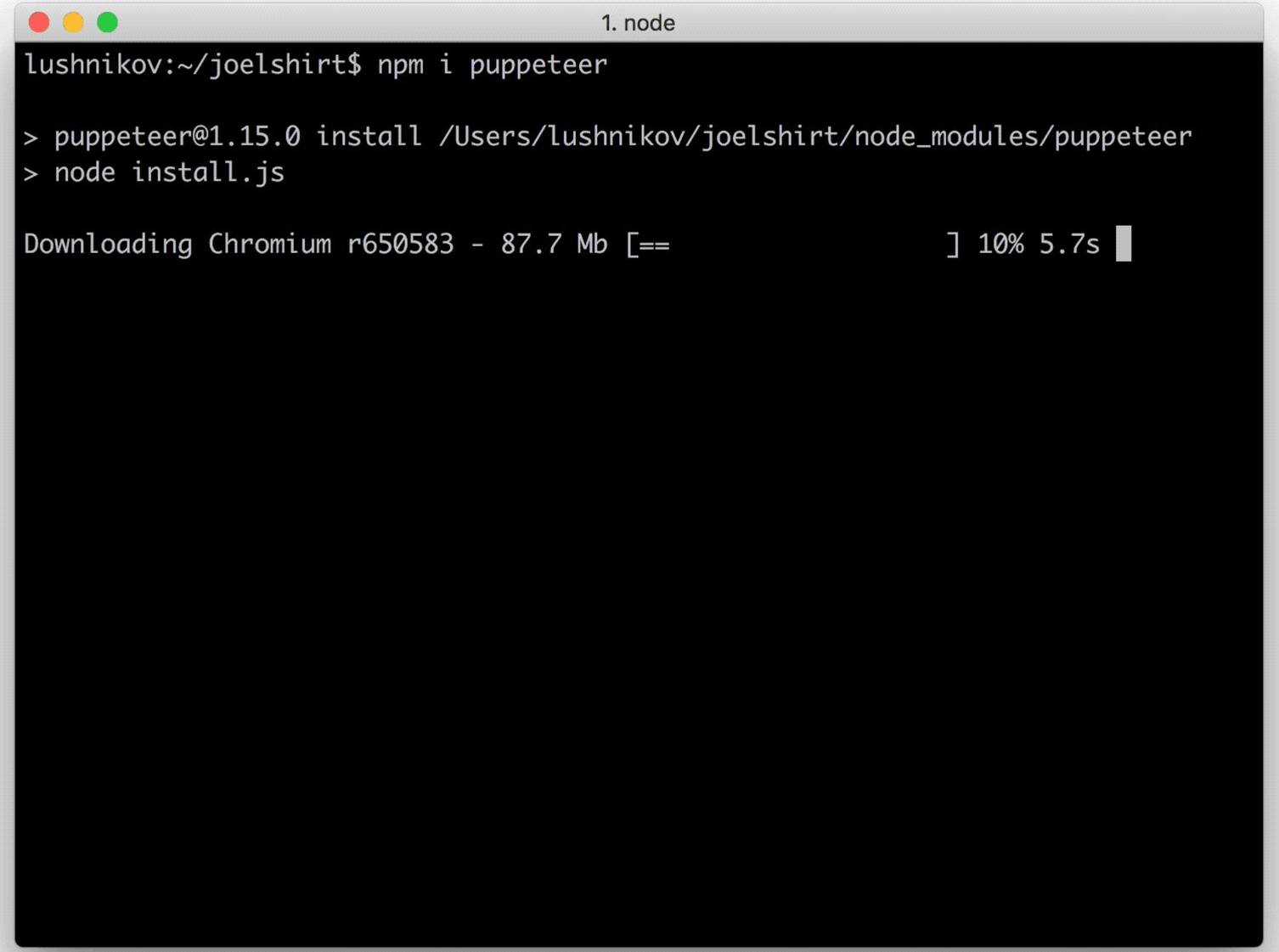
HEISENBUG



HEISENBUG

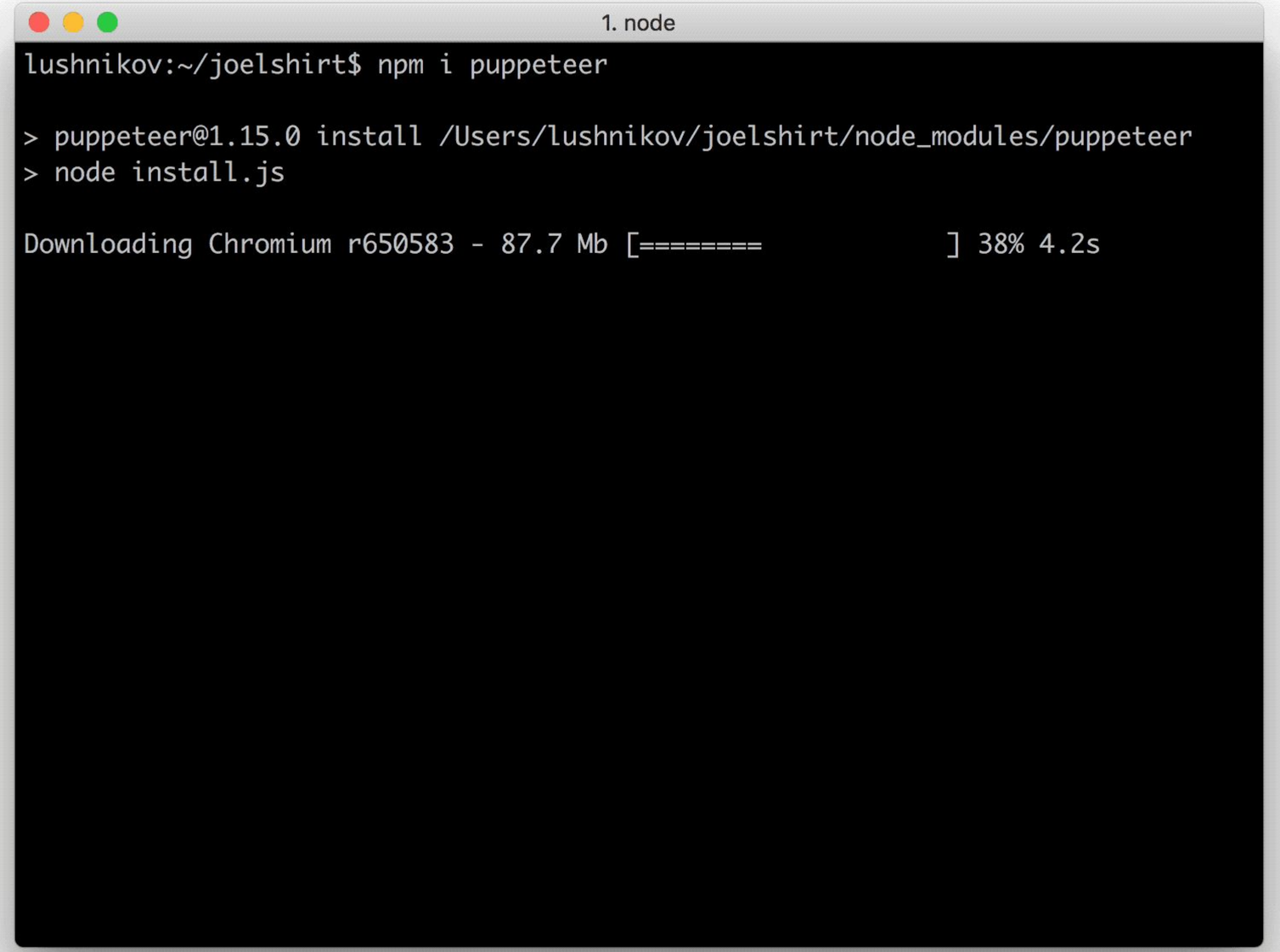


HEISENBUG


1. node
lushnikov:~/joelshirt$ npm i puppeteer
> puppeteer@1.15.0 install /Users/lushnikov/joelshirt/node_modules/puppeteer
> node install.js
Downloading Chromium r650583 - 87.7 Mb [==   ] 10% 5.7s

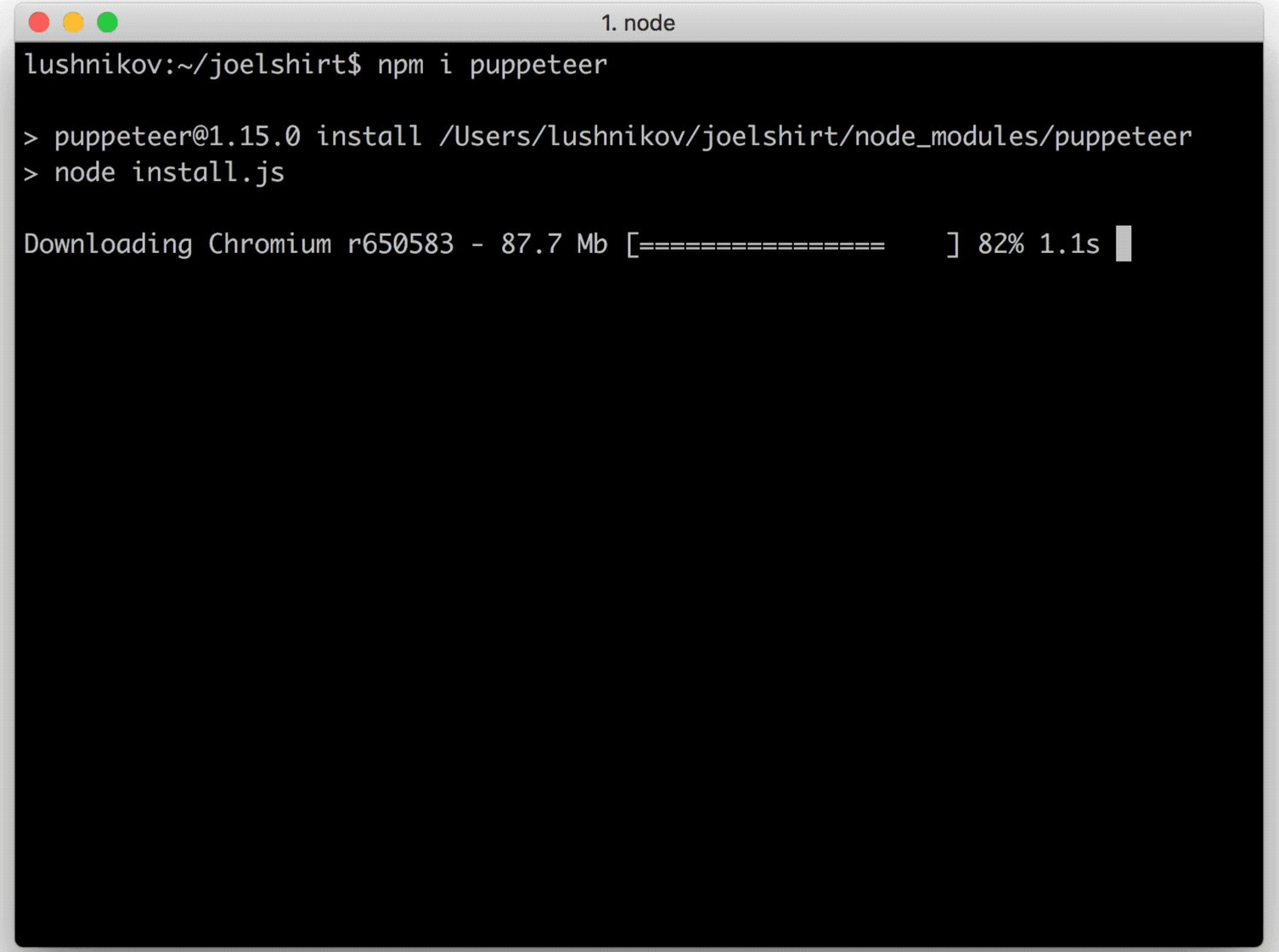

Puppeteer

HEISENBUG


1. node
lushnikov:~/joelshirt$ npm i puppeteer
> puppeteer@1.15.0 install /Users/lushnikov/joelshirt/node_modules/puppeteer
> node install.js
Downloading Chromium r650583 - 87.7 Mb [========   ] 38% 4.2s

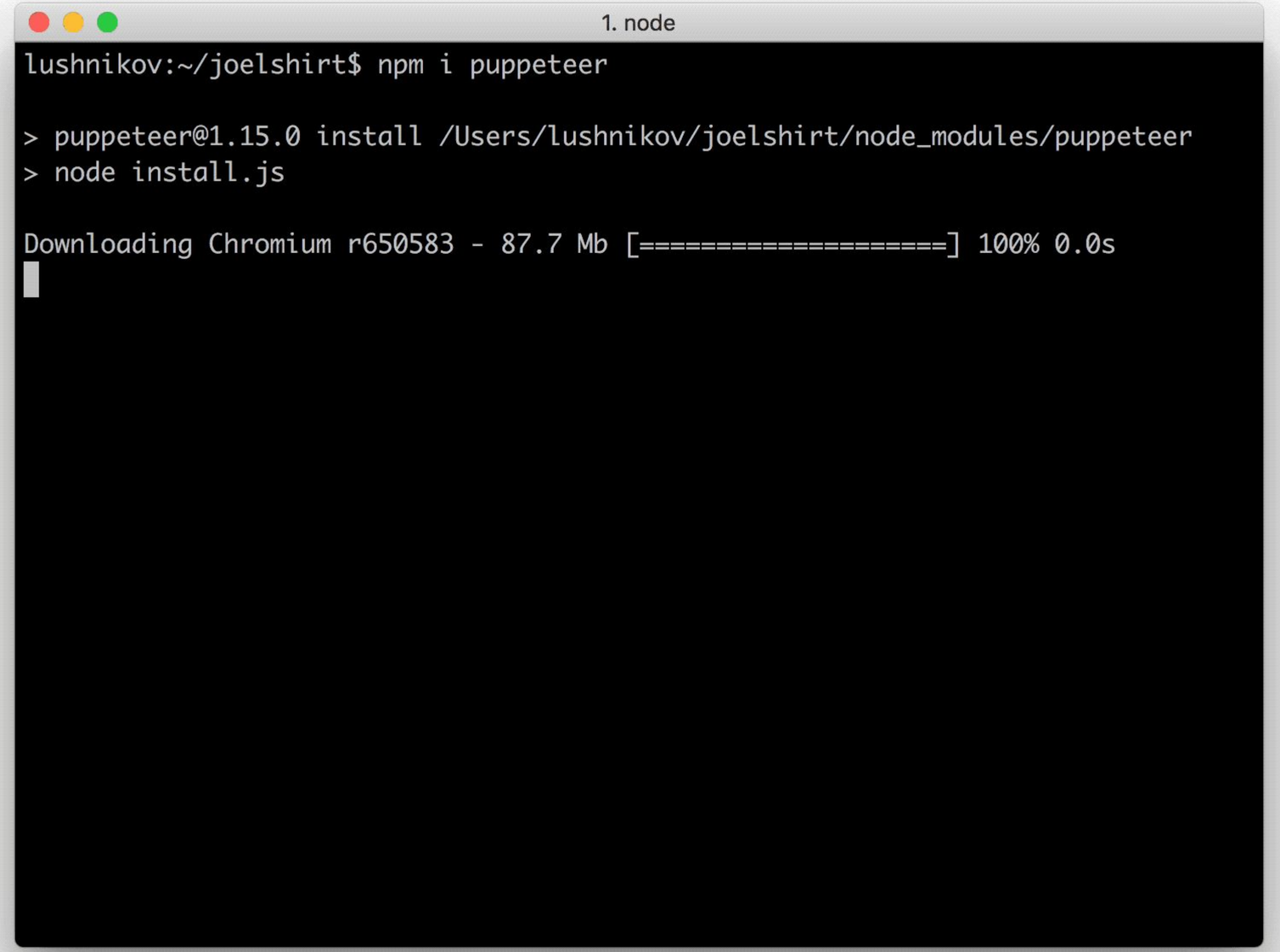

Puppeteer

HEISENBUG


1. node
lushnikov:~/joelshirt$ npm i puppeteer
> puppeteer@1.15.0 install /Users/lushnikov/joelshirt/node_modules/puppeteer
> node install.js
Downloading Chromium r650583 - 87.7 Mb [================   ] 82% 1.1s

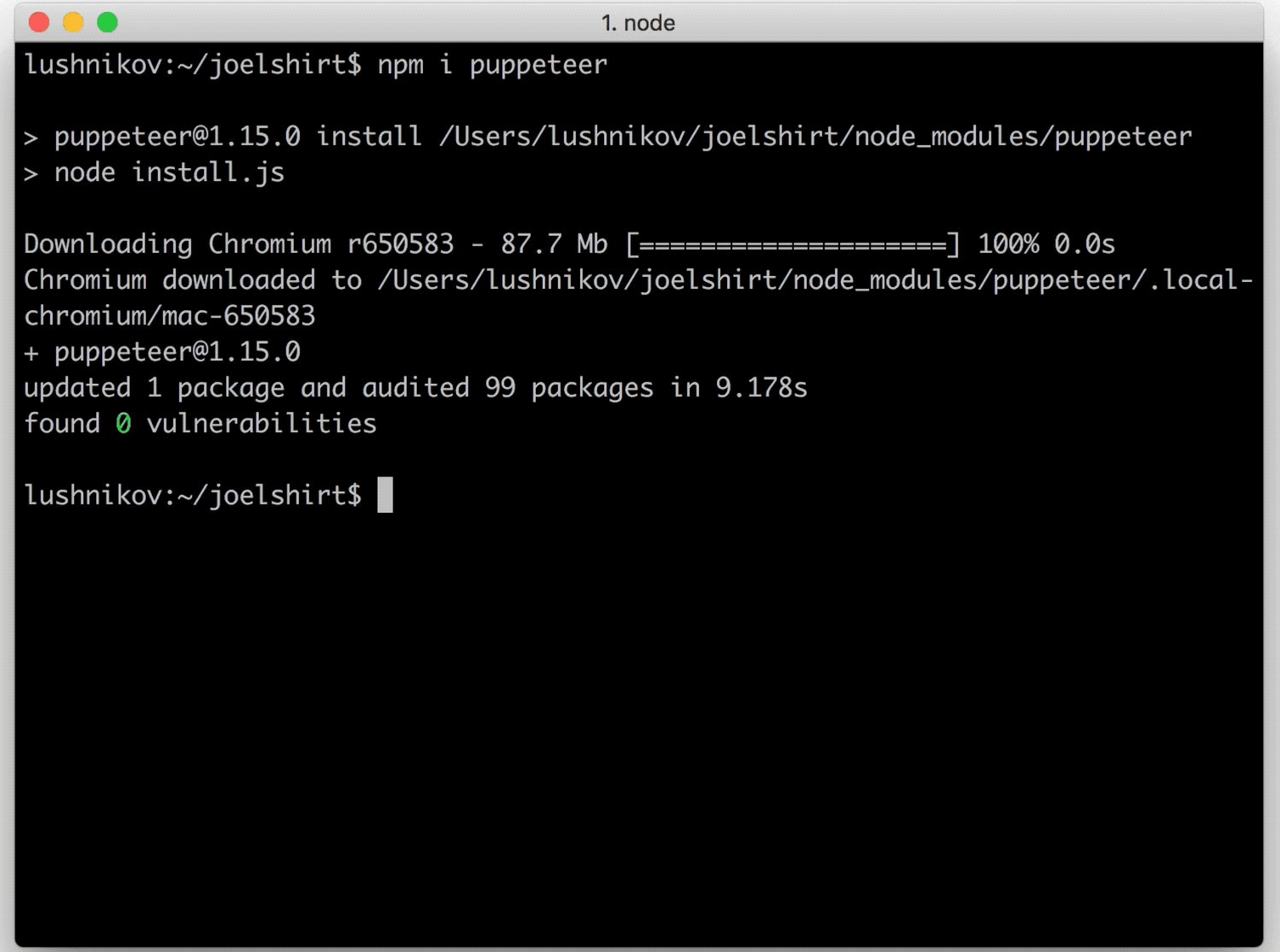

Puppeteer

HEISENBUG


1. node
lushnikov:~/joelshirt$ npm i puppeteer
> puppeteer@1.15.0 install /Users/lushnikov/joelshirt/node_modules/puppeteer
> node install.js
Downloading Chromium r650583 - 87.7 Mb [====================] 100% 0.0s


Puppeteer

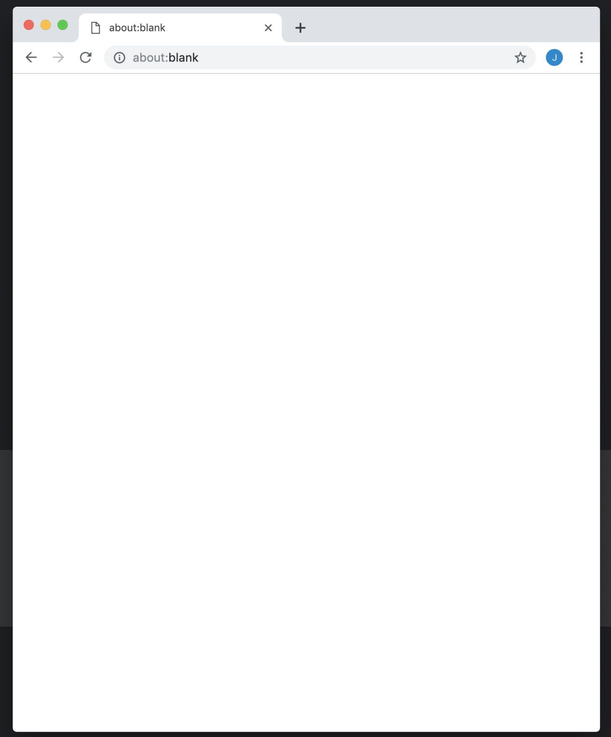
HEISENBUG


1. node
lushnikov:~/joelshirt$ npm i puppeteer
> puppeteer@1.15.0 install /Users/lushnikov/joelshirt/node_modules/puppeteer
> node install.js
Downloading Chromium r650583 - 87.7 Mb [====================] 100% 0.0s
Chromium downloaded to /Users/lushnikov/joelshirt/node_modules/puppeteer/.local-chromium/mac-650583
+ puppeteer@1.15.0
updated 1 package and audited 99 packages in 9.178s
found 0 vulnerabilities
lushnikov:~/joelshirt$

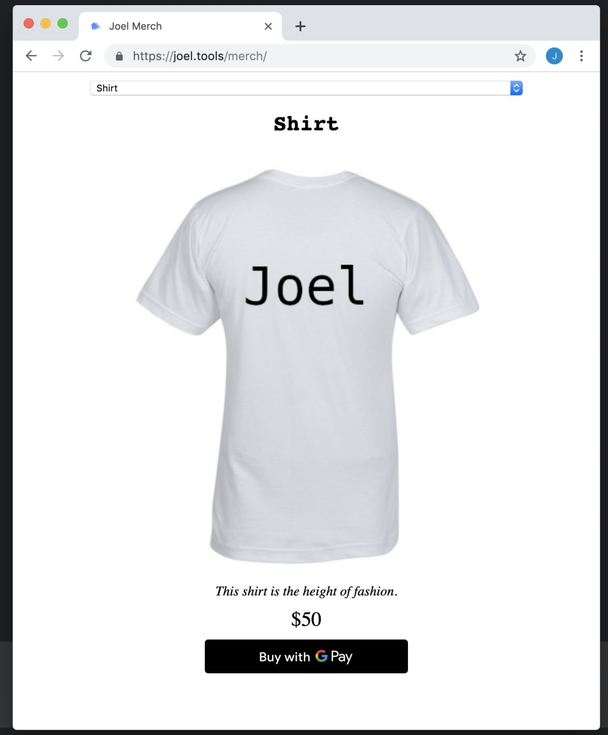

Puppeteer

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

(async () => {
  // Automate here




})();
```

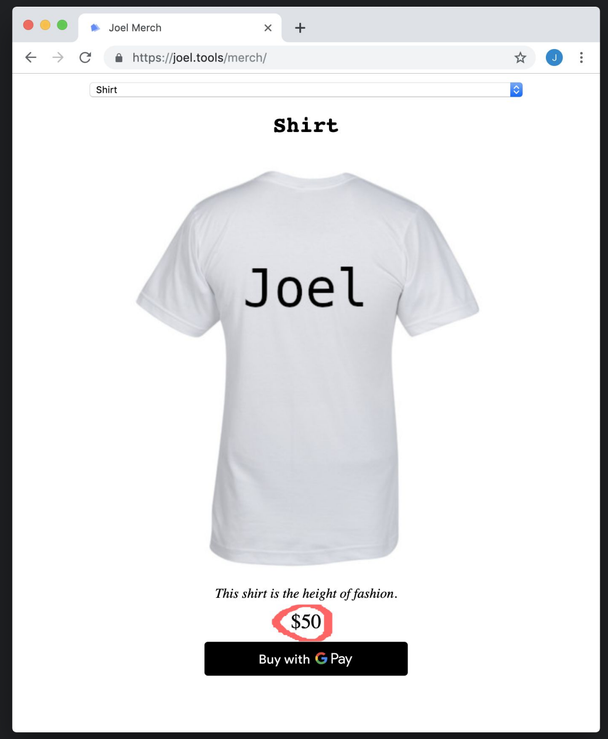
```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();




})();
```



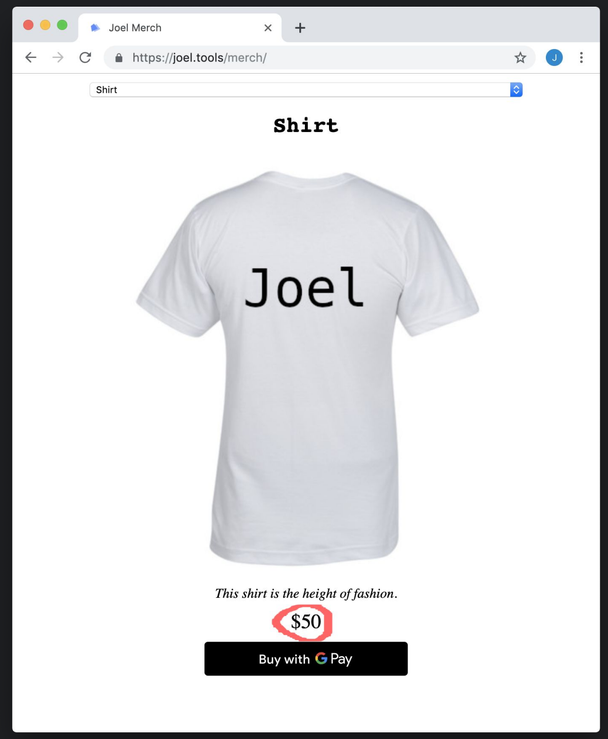
```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');



})();
```



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

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');
  const price = await page.$eval('.price', div => div.textContent);


})();
```



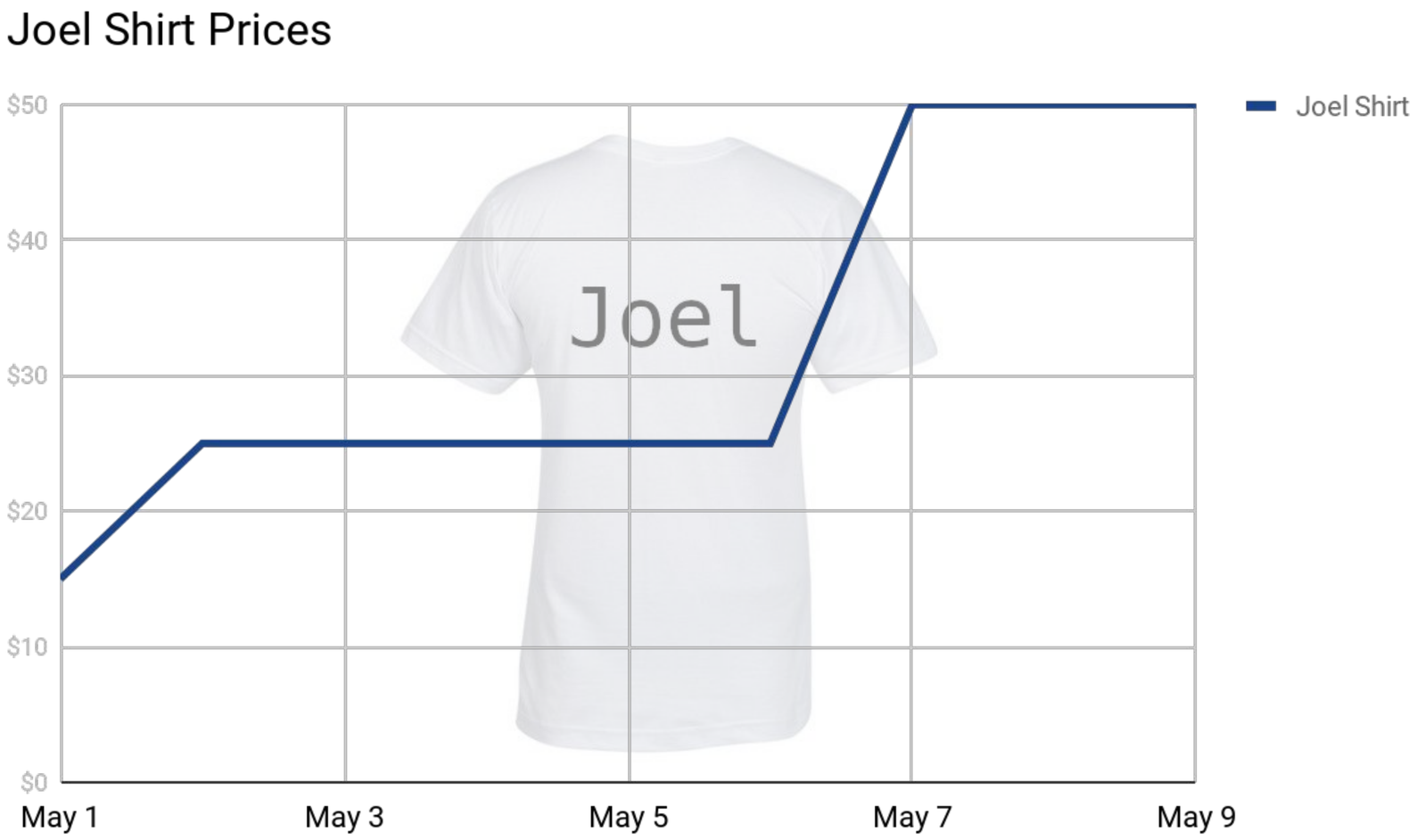
```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');
  const price = await page.$eval('.price', div => div.textContent);
  console.log(price);

})();
```



$50

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

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');
  const price = await page.$eval('.price', div => div.textContent);
  console.log(price);
  await browser.close();
})();
```

$50



Puppeteer

HEISENBUG

Puppeteer for Firefox

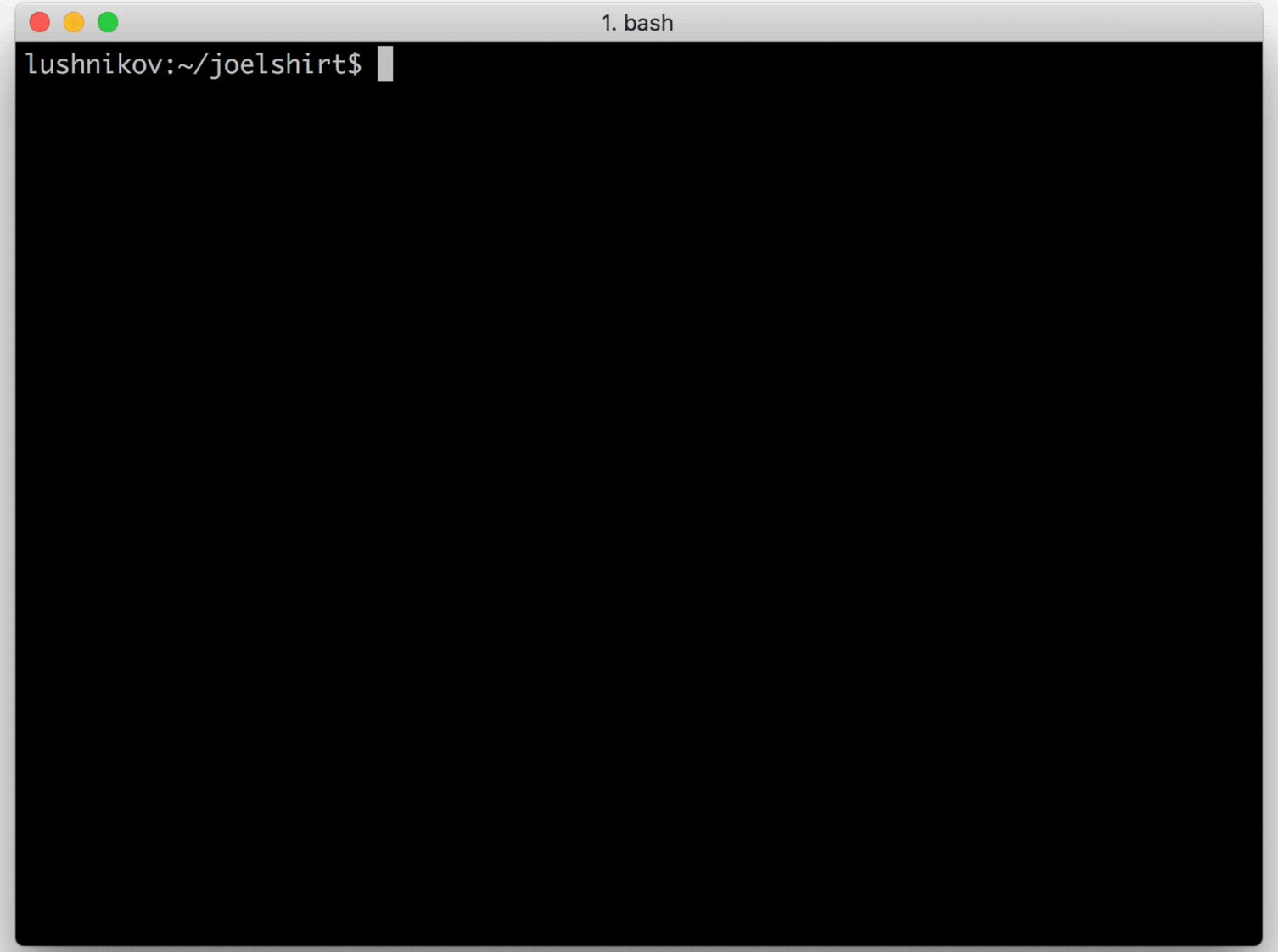
puppeteer-firefox

🐊 **experimental** project

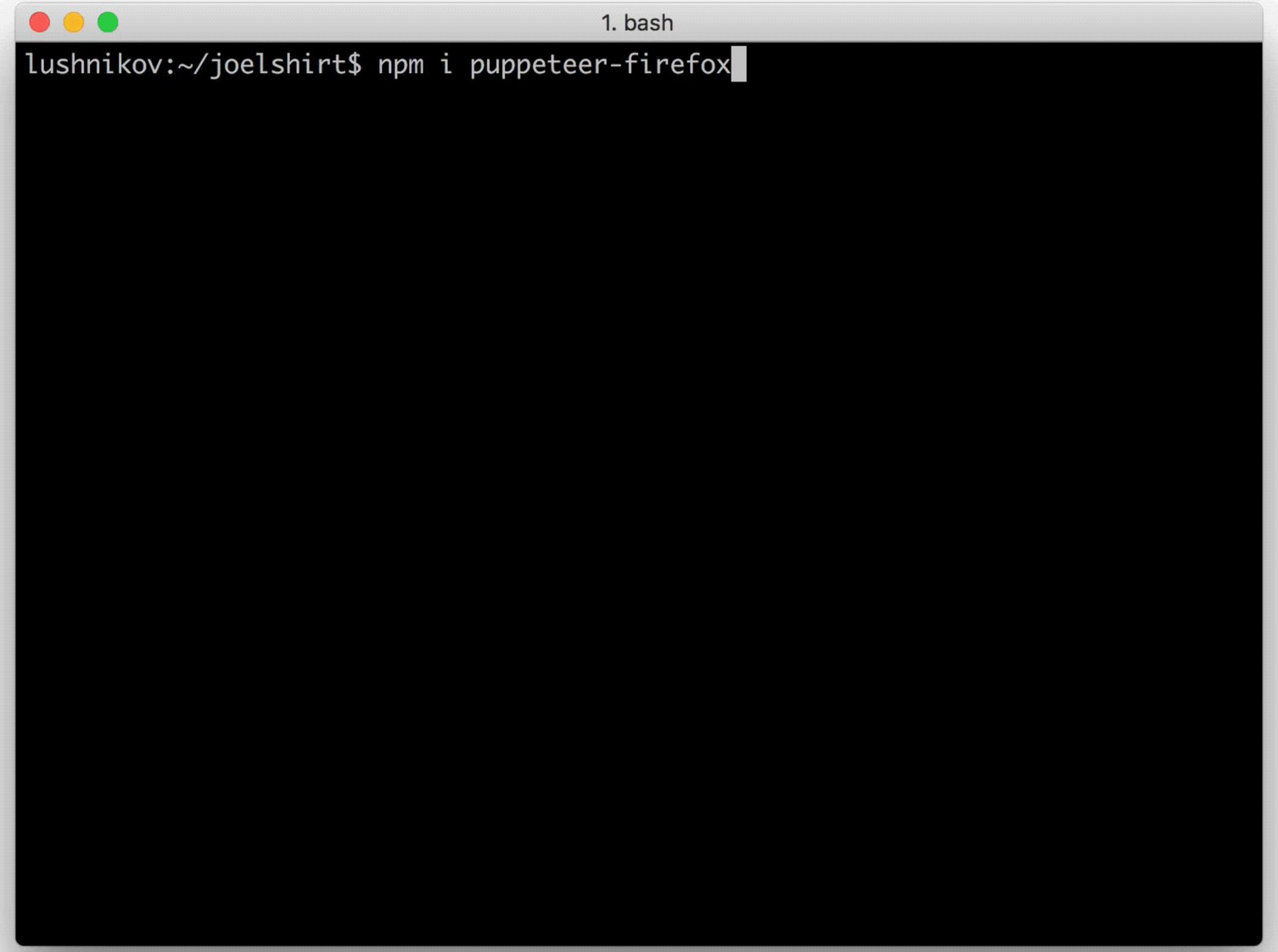
👉 **`npm i puppeteer-firefox`**

👉 supports **90%** of Puppeteer API

HEISENBUG

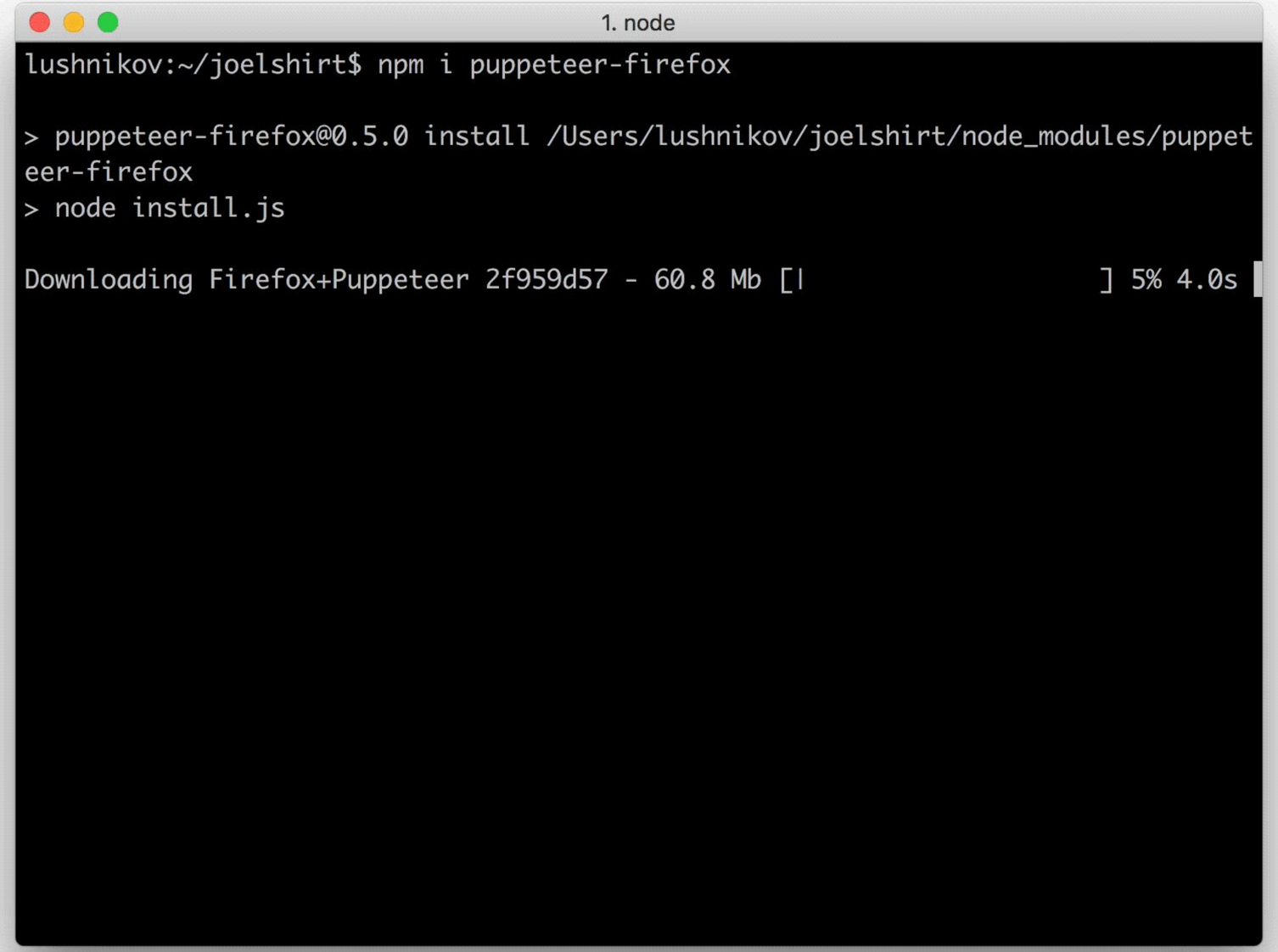


HEISENBUG


1. bash
lushnikov:~/joelshirt$ npm i puppeteer-firefox

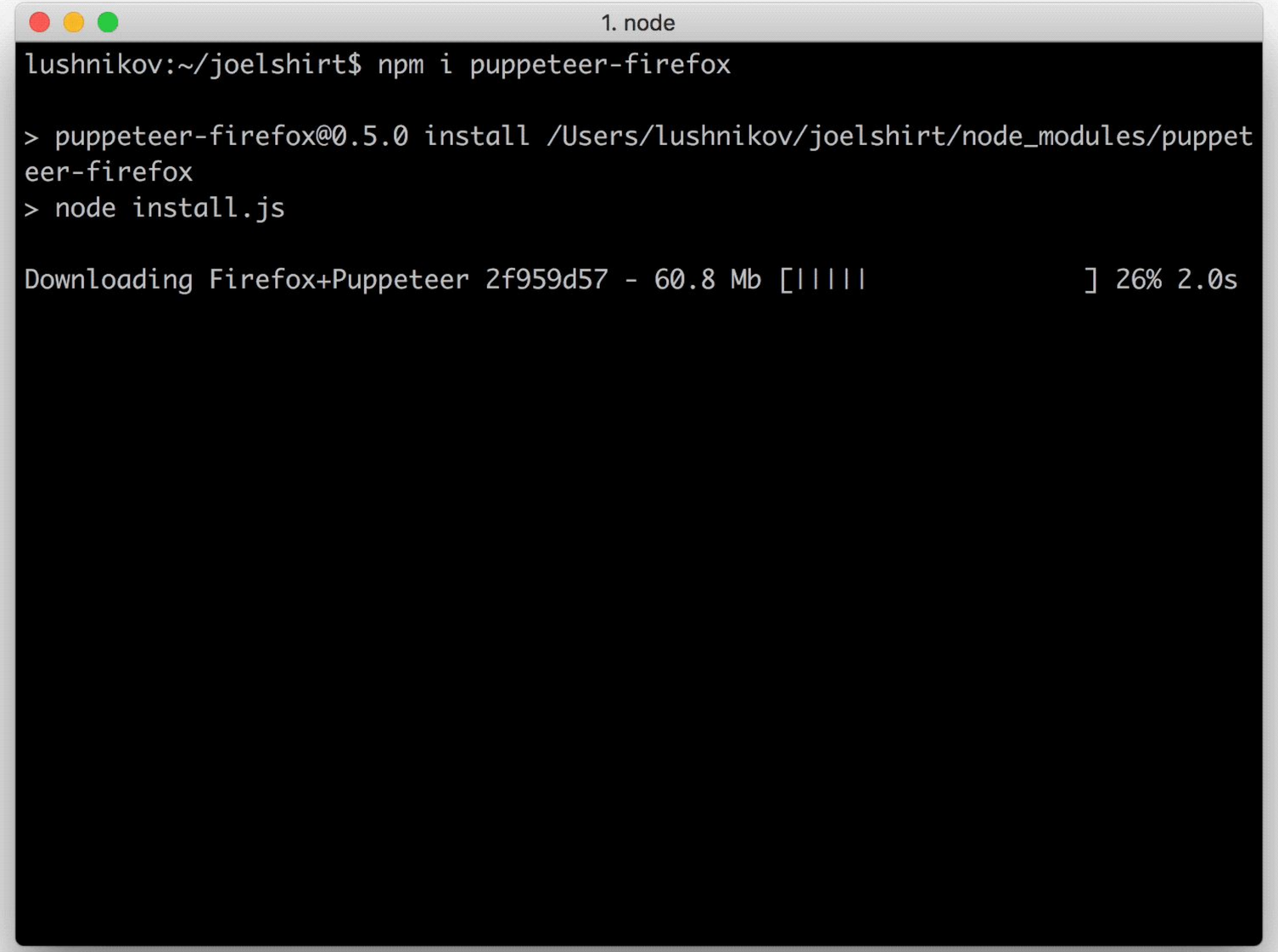

Puppeteer

HEISENBUG


1. node
lushnikov:~/joelshirt$ npm i puppeteer-firefox
> puppeteer-firefox@0.5.0 install /Users/lushnikov/joelshirt/node_modules/puppeteer-firefox
> node install.js
Downloading Firefox+Puppeteer 2f959d57 - 60.8 Mb [|] 5% 4.0s

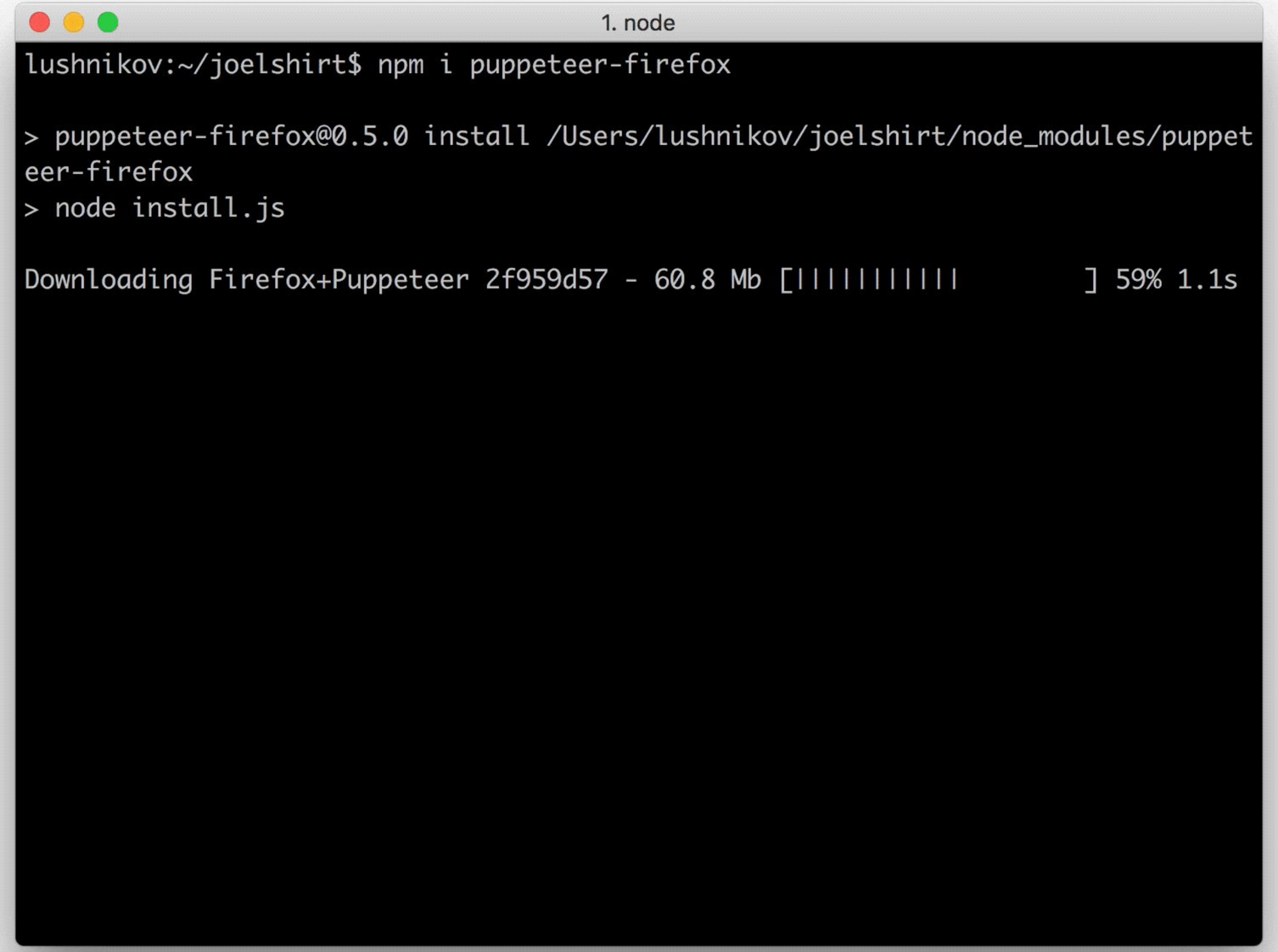

Puppeteer

HEISENBUG


1. node
lushnikov:~/joelshirt$ npm i puppeteer-firefox
> puppeteer-firefox@0.5.0 install /Users/lushnikov/joelshirt/node_modules/puppeteer-firefox
> node install.js
Downloading Firefox+Puppeteer 2f959d57 - 60.8 Mb [|||||] 26% 2.0s

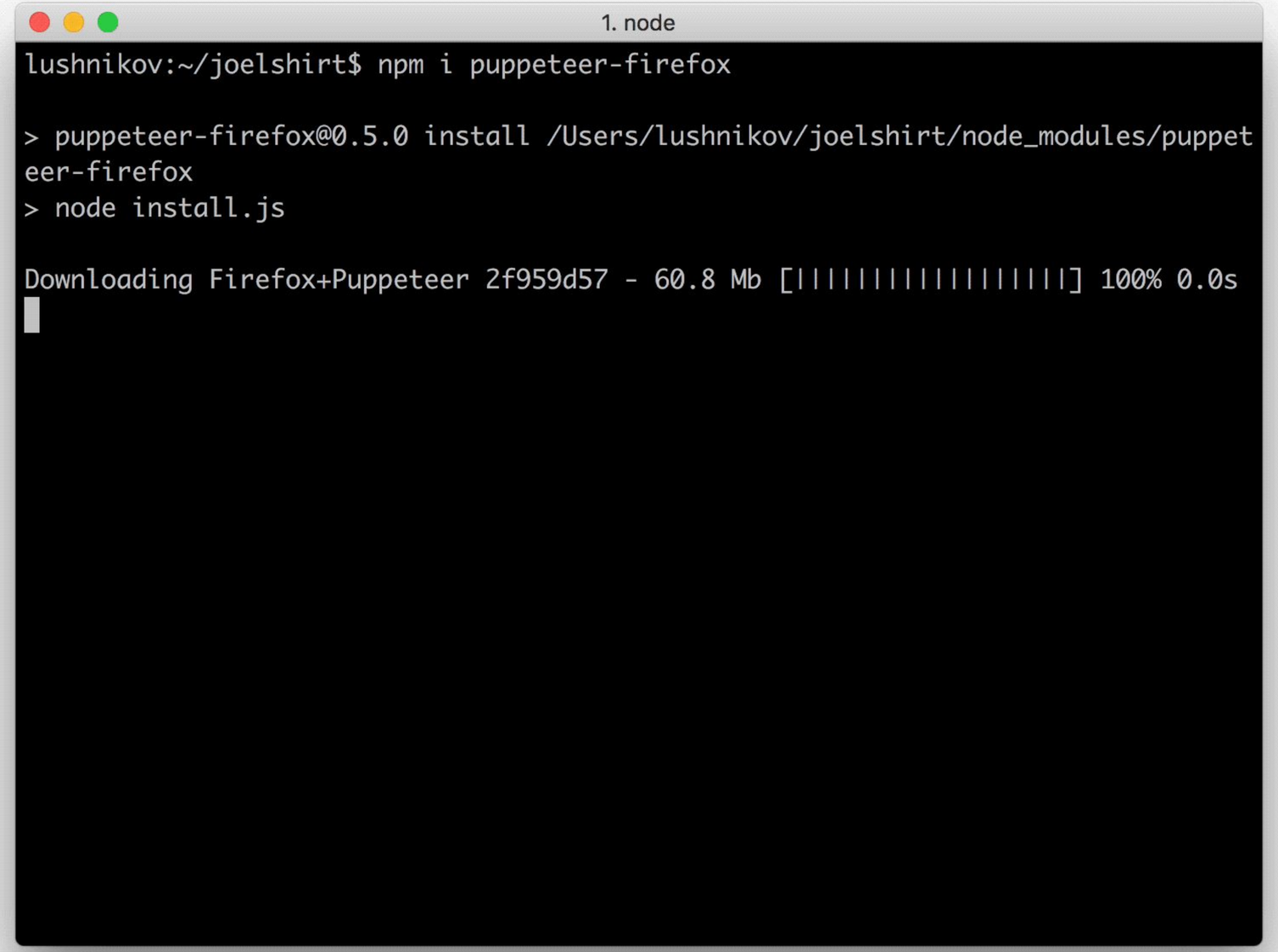

Puppeteer

HEISENBUG


1. node
lushnikov:~/joelshirt$ npm i puppeteer-firefox
> puppeteer-firefox@0.5.0 install /Users/lushnikov/joelshirt/node_modules/puppeteer-firefox
> node install.js
Downloading Firefox+Puppeteer 2f959d57 - 60.8 Mb [|||||||||||] 59% 1.1s

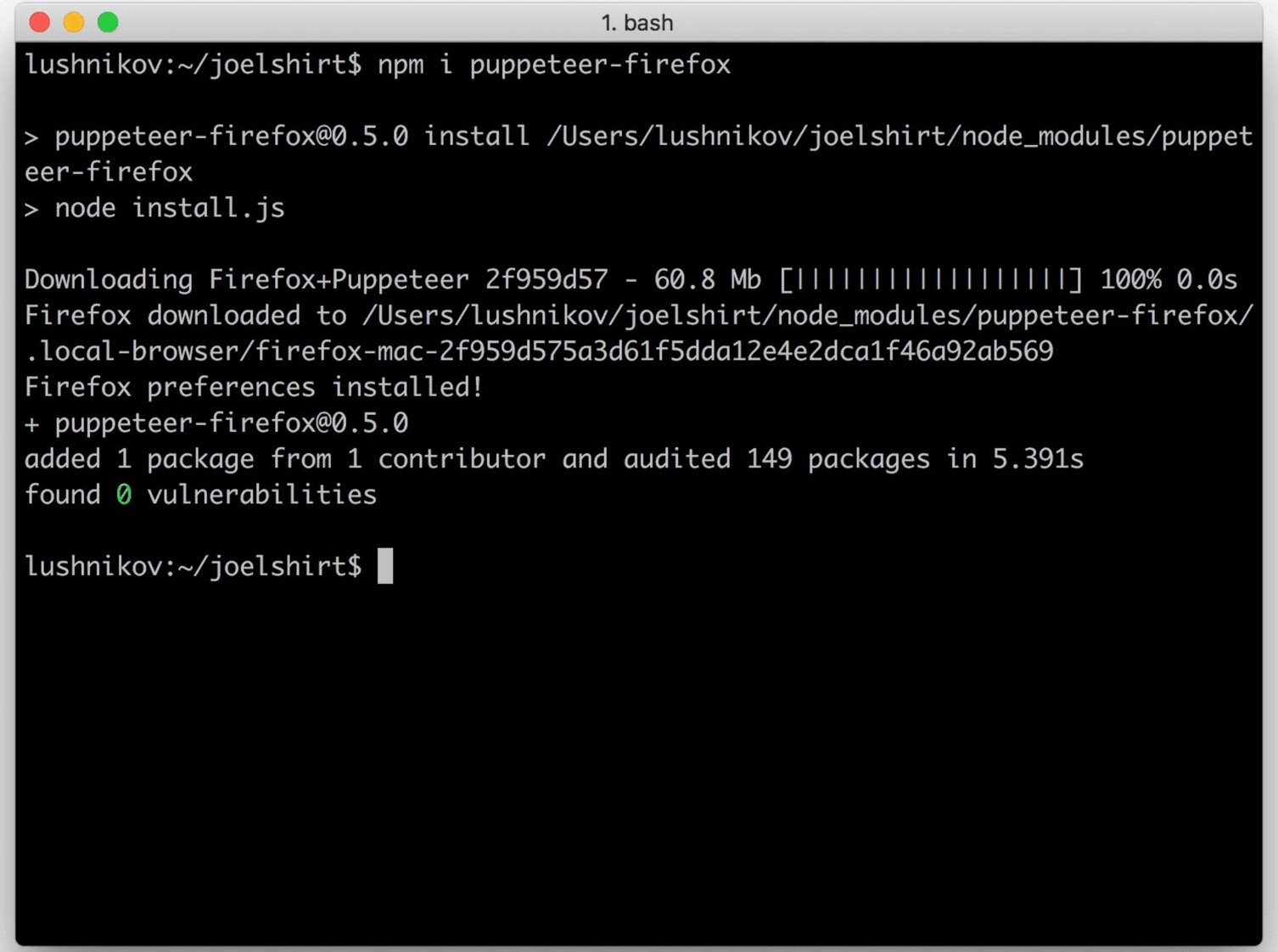

Puppeteer

HEISENBUG


1. node
lushnikov:~/joelshirt$ npm i puppeteer-firefox
> puppeteer-firefox@0.5.0 install /Users/lushnikov/joelshirt/node_modules/puppeteer-firefox
> node install.js
Downloading Firefox+Puppeteer 2f959d57 - 60.8 Mb [||||||||||||||||||] 100% 0.0s


Puppeteer

HEISENBUG

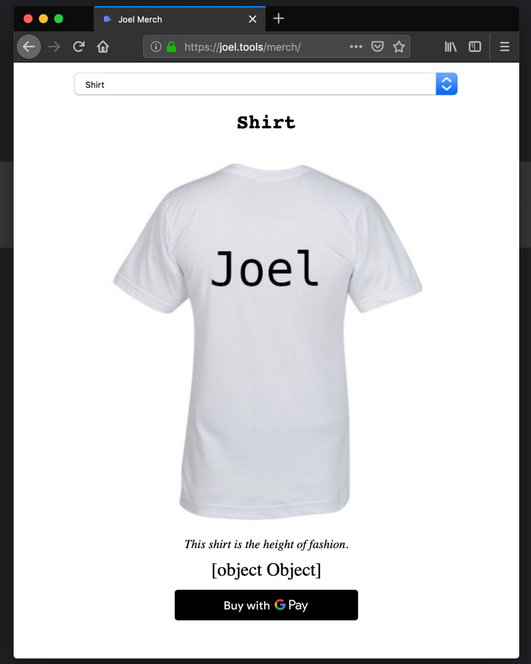

1. bash
lushnikov:~/joelshirt$ npm i puppeteer-firefox
> puppeteer-firefox@0.5.0 install /Users/lushnikov/joelshirt/node_modules/puppeteer-firefox
> node install.js
Downloading Firefox+Puppeteer 2f959d57 - 60.8 Mb [|||||||||||||||||] 100% 0.0s
Firefox downloaded to /Users/lushnikov/joelshirt/node_modules/puppeteer-firefox/.local-browser/firefox-mac-2f959d575a3d61f5dda12e4e2dca1f46a92ab569
Firefox preferences installed!
+ puppeteer-firefox@0.5.0
added 1 package from 1 contributor and audited 149 packages in 5.391s
found 0 vulnerabilities
lushnikov:~/joelshirt$


Puppeteer

HEISENBUG

```
const puppeteer = require('puppeteer-firefox');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');
  const price = await page.$eval('.price', div => div.textContent);
  console.log(price);
  await browser.close();
})();
```

```
const puppeteer = require('puppeteer-firefox');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');
  const price = await page.$eval('.price', div => div.textContent);
  console.log(price);
  await browser.close();
})();
```



[object Object]

[object Object]

Web Testing Challenges

👉 web testing is **complicated**
👉 web testing is **slow**
👉 web testing is **flaky**

HEISENBUG

New Web Testing

Powered by Puppeteer

Simple

~~Complicated~~ Setup

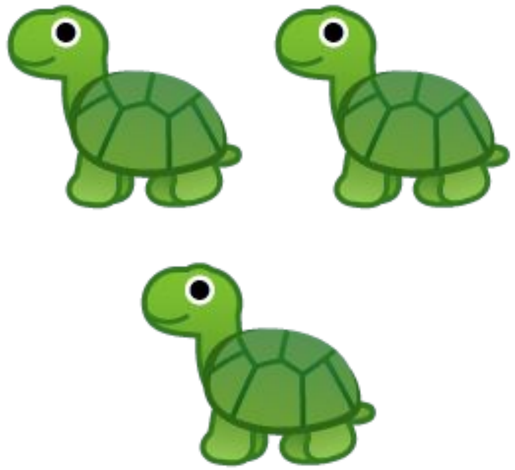
👉 same thing as any NPM library

👉 any test runner: jest, ava, mocha...

Runs Everywhere

- ✔️ **Desktop**: Mac, Windows, Linux
- ✔️ **CI**: Travis-CI, AppVeyor, ...
- ✔️ **Cloud**: GCP, AWS, Azure, ...
- ✔️ **Docker Containers**

Tests are Slow



HEISENBUG

Joel Merch

https://joel.tools/merch/

Shirt

Shirt

This shirt is the height of fashion.

$50

Buy with G Pay

Tests are ~~Slow~~ Fast

👉 Browser Contexts

100x faster

```
it('should have a pay button', async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');
  expect(await page.$('button.gpay-button')).toBeTruthy();
  await browser.close();
})
```

```
const browser = await puppeteer.launch();
...
it('should have a pay button', async () => {
  const context = await browser.createIncognitoBrowserContext();
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch');
  expect(await page.$('button.gpay-button')).toBeTruthy();
  await context.close();
})
```

Tests are Flaky

HEISENBUG

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');

  await page.click('button.gpay-button');

  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitFor(1000);
  await page.click('button.gpay-button');
  await page.waitFor(1000);
  assert(await page.$('.successful-purchase'));
});
```

Tests are ~~Flaky~~ reliable

- 👉 `page.waitForSelector()`
- 👉 `page.waitForRequest()`
- 👉 `page.waitForResponse()`
- 👉 `page.waitForFunction()`
- 👉 `page.waitForNavigation()`

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitFor(1000);
  await page.click('button.gpay-button');
  await page.waitFor(1000);
  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitFor(1000);
  await page.click('button.gpay-button');
  await page.waitFor(1000);
  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitForSelector('button.gpay-button');
  await page.click('button.gpay-button');
  await page.waitFor(1000);
  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitForSelector('button.gpay-button');
  await page.click('button.gpay-button');
  await page.waitFor(1000);
  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitForSelector('button.gpay-button');


  await page.click('button.gpay-button');
  await page.waitFor(1000);

  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitForSelector('button.gpay-button');


  await page.click('button.gpay-button');
  await page.waitFor(1000);

  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitForSelector('button.gpay-button');


  await page.click('button.gpay-button');


  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitForSelector('button.gpay-button');


  await page.click('button.gpay-button');


  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitForSelector('button.gpay-button');

  const response = page.waitForResponse(res => res.url().endsWith('/pay'));
  await page.click('button.gpay-button');


  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitForSelector('button.gpay-button');

  const response = page.waitForResponse(res => res.url().endsWith('/pay'));
  await page.click('button.gpay-button');


  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitForSelector('button.gpay-button');

  const response = page.waitForResponse(res => res.url().endsWith('/pay'));
  await page.click('button.gpay-button');
  await response;

  assert(await page.$('.successful-purchase'));
});
```

```
it('should pay', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch/');
  await page.waitForSelector('button.gpay-button');

  await Promise.all([
    page.waitForResponse(res => res.url().endsWith('/pay')),
    page.click('button.gpay-button'),
  ]);
  assert(await page.$('.successful-purchase'));
});
```

Recap

👉 Testing with Puppeteer is **fast**

👉 Testing with Puppeteer is **reliable**

👉 Testing with Puppeteer is **simple**


Puppeteer

Testing Fundamentals

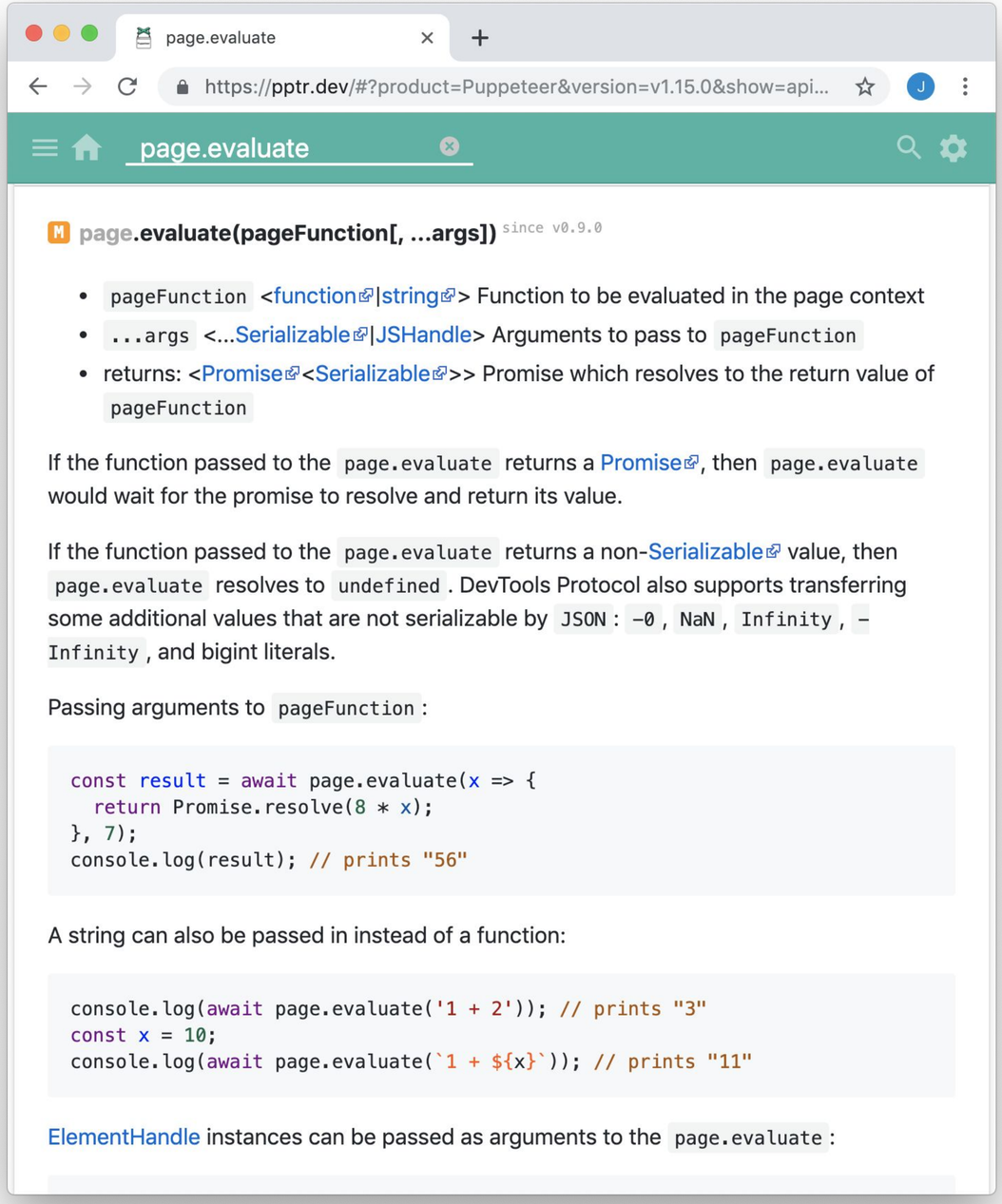
Web Testing Is...

👉 Navigation

👉 Evaluation

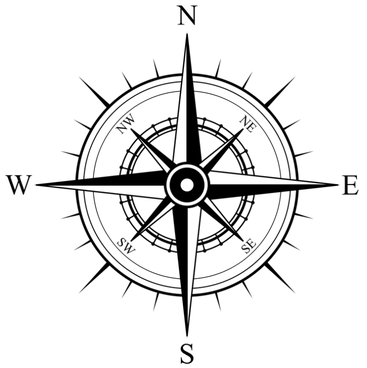
👉 Input Injection

pptr.dev



HEISENBUG

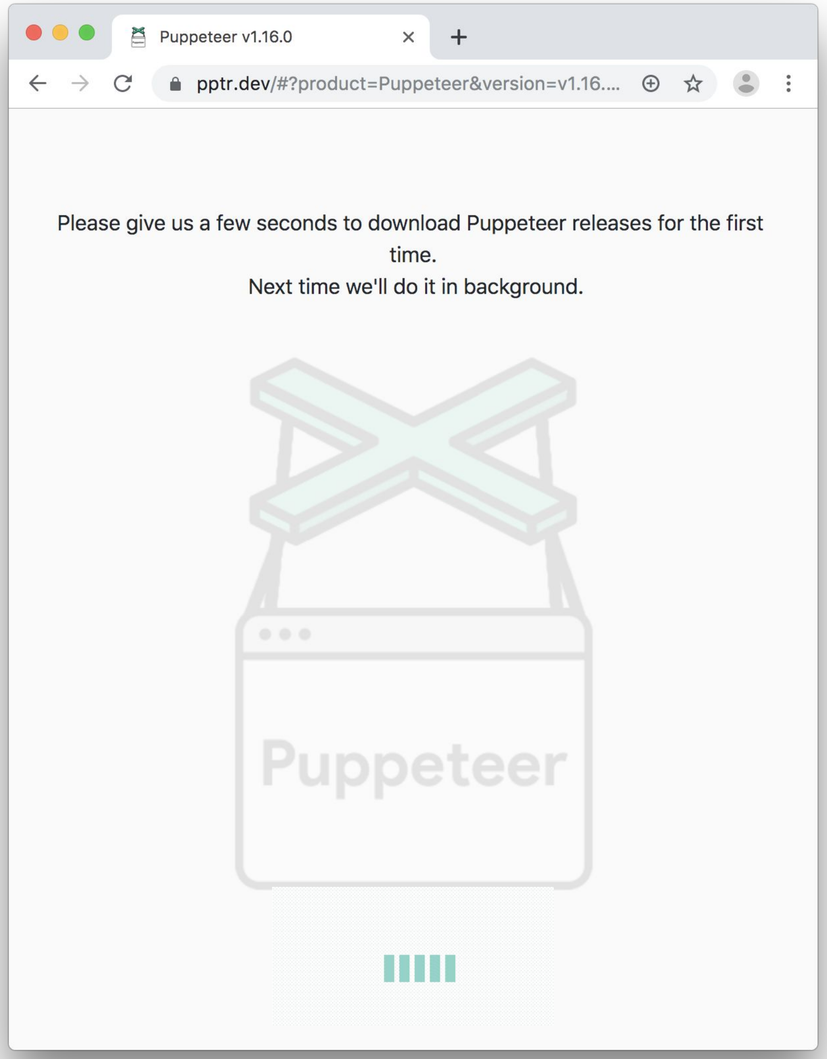
Navigation




```
👉 await page.goto('https://pptr.dev');
```

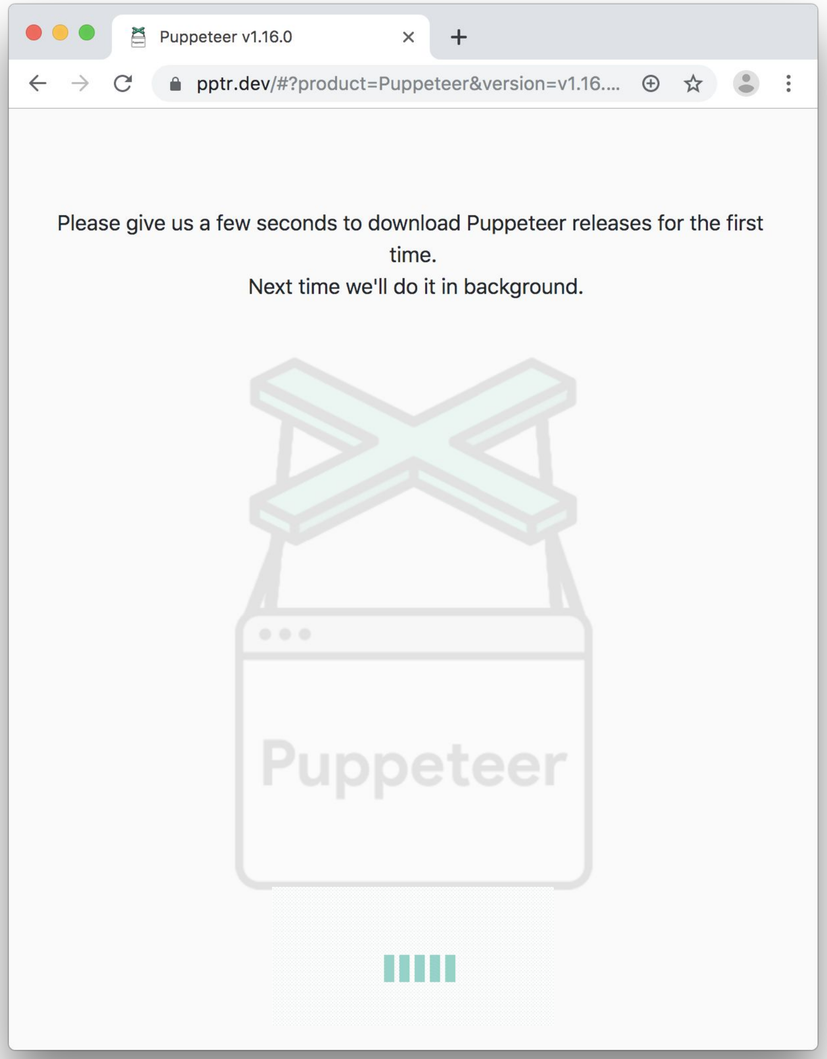
👉 resolves when page emits “**load**” event

```
await page.goto('https://pptr.dev');
```



HEISENBUG

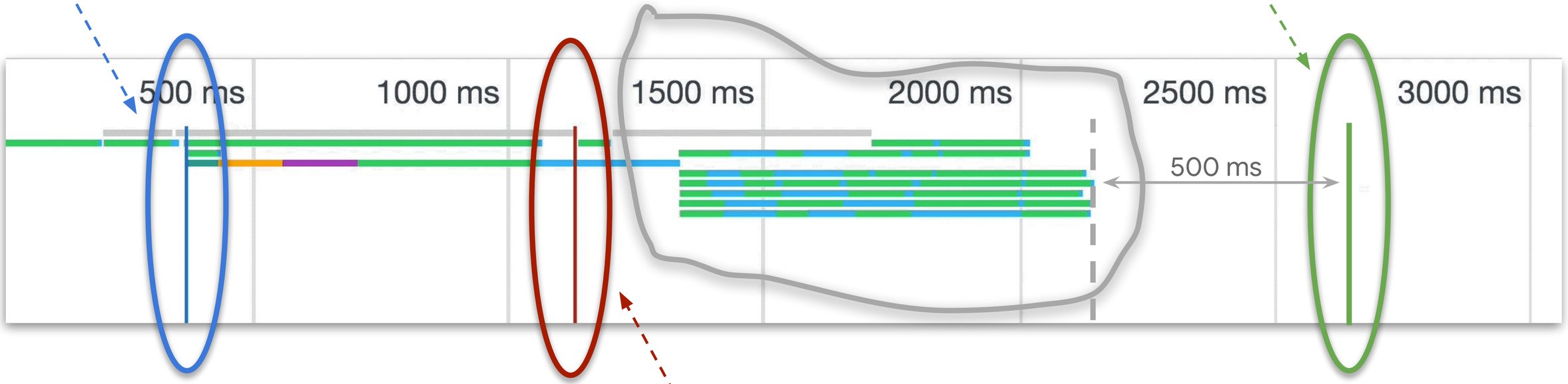
```
await page.goto('https://pptr.dev', {
  waitUntil: '???'
});
```



HEISENBUG

```
await page.goto('https://pptr.dev', {
  waitUntil: 'domcontentloaded'
});
```

```
await page.goto('https://pptr.dev', {
  waitUntil: 'networkidle0'
});
```

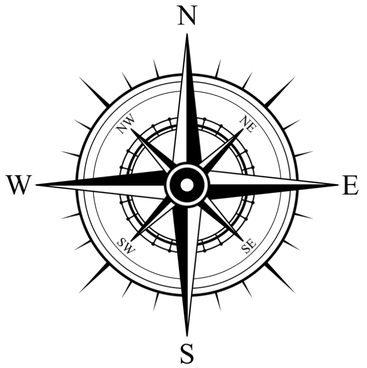


DOMContentLoaded **load** **networkidle0**

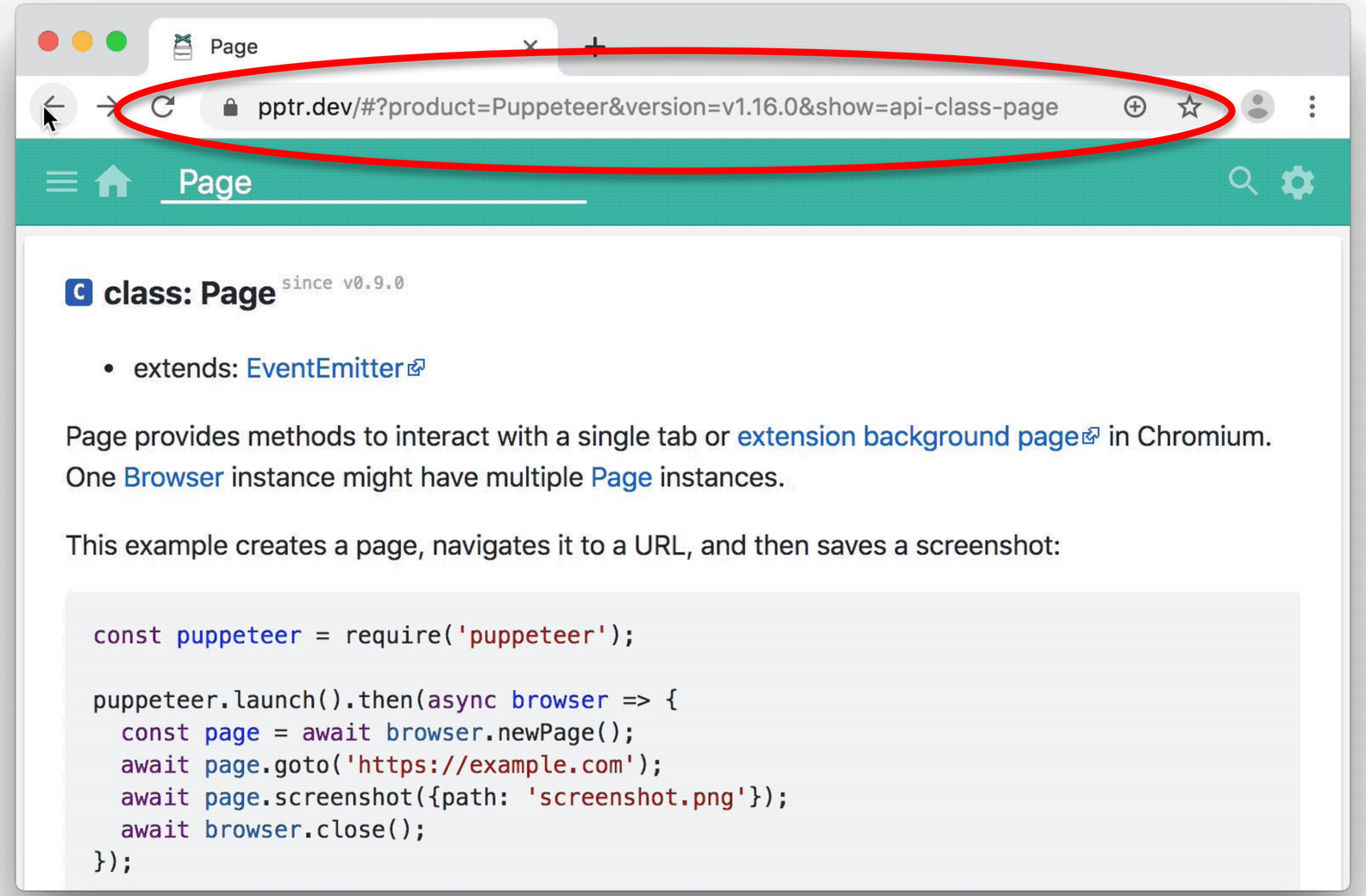
```
await page.goto('https://pptr.dev');
```

HEISENBUG

Clicking Links

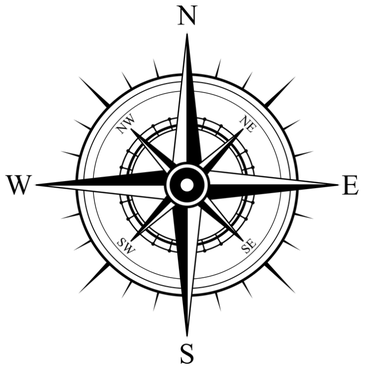


- 👉 navigate to a different website
- 👉 anchor navigation
- 👉 Single-Page navigation: History API


Page
pptr.dev/#?product=Puppeteer&version=v1.16.0&show=api-class-page
Page
class: Page since v0.9.0
extends: EventEmitter
Page provides methods to interact with a single tab or extension background page in Chromium. One Browser instance might have multiple Page instances.
This example creates a page, navigates it to a URL, and then saves a screenshot:
const puppeteer = require('puppeteer');
puppeteer.launch().then(async browser => {
const page = await browser.newPage();
await page.goto('https://example.com');
await page.screenshot({path: 'screenshot.png'});
await browser.close();
});


Puppeteer

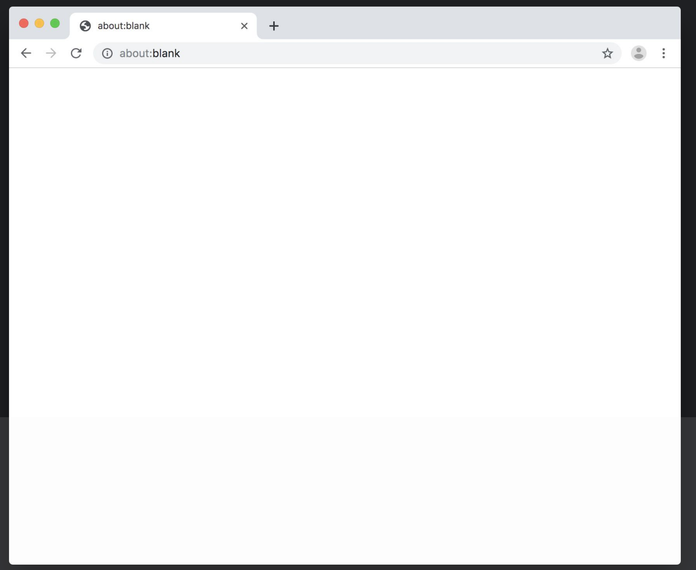
Navigation Tracking



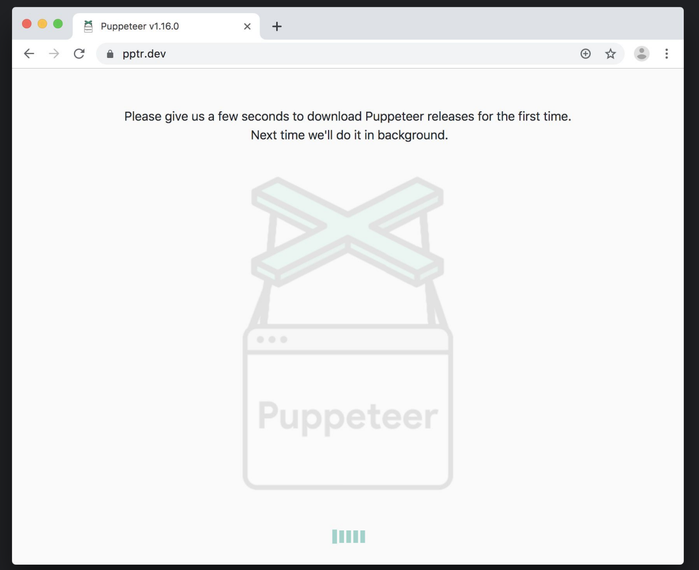
👉 `page.waitForNavigation();`

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

(async () => {
  // Automate here




})();
```

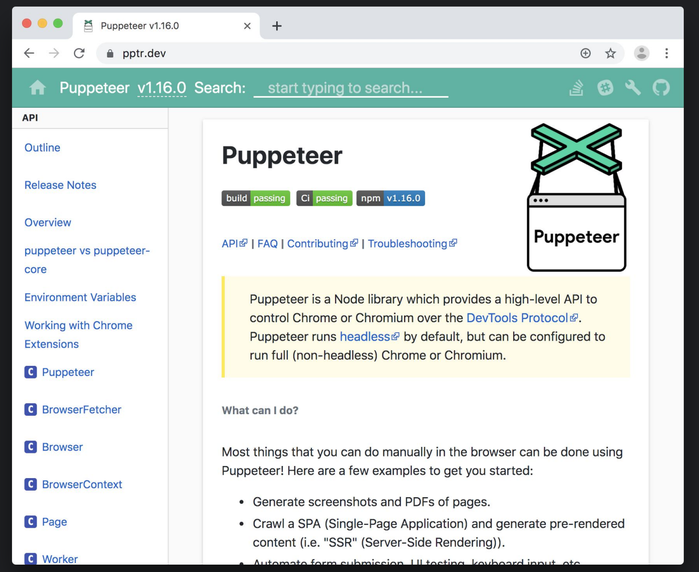
```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();




})();
```



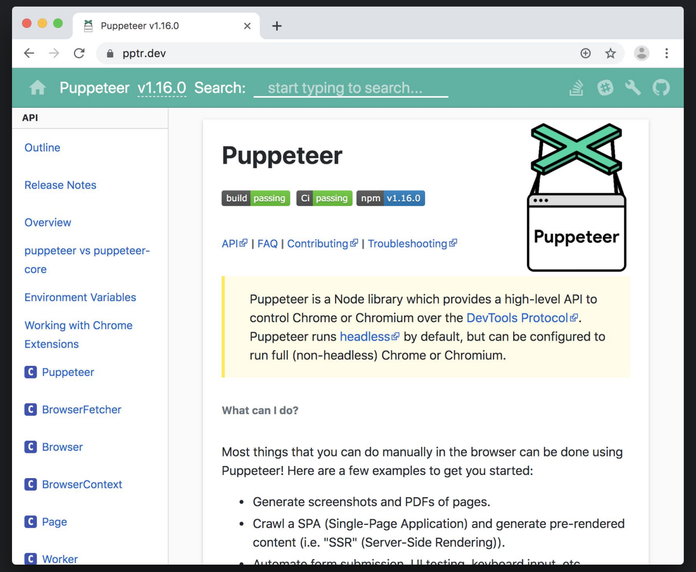
```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev');



})();
```



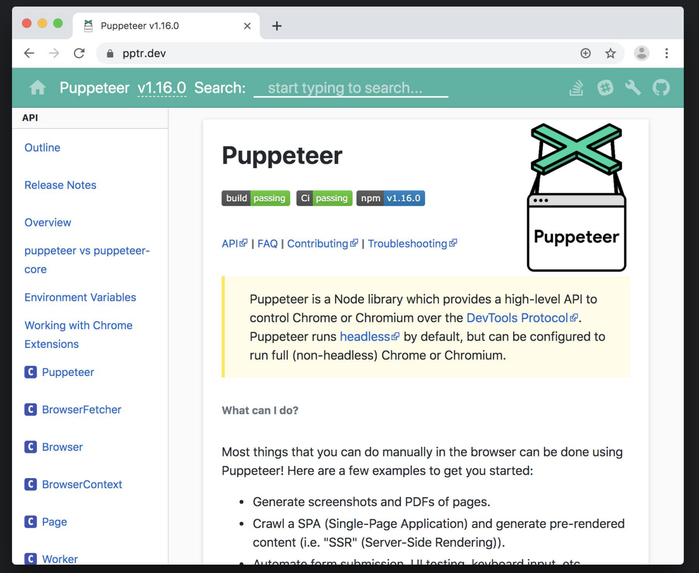
```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev', {waitUntil: 'networkidle0'});

})();
```



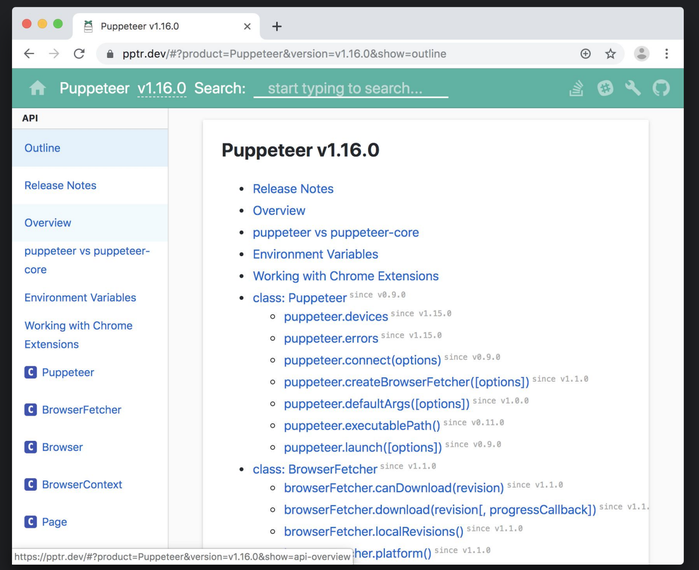
```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev', {waitUntil: 'networkidle0'});
  await Promise.all([



  ]);
})();
```



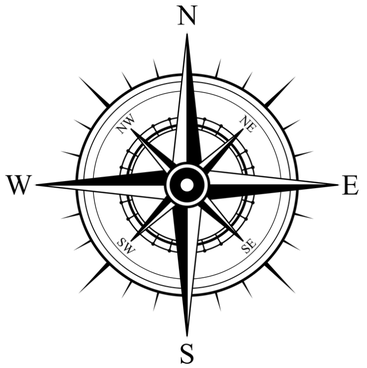
```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev', {waitUntil: 'networkidle0'});
  await Promise.all([
    page.click('sidebar-component a'), // Documentation Outline

  ]);
})();
```



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

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev', {waitUntil: 'networkidle0'});
  await Promise.all([
    page.click('sidebar-component a'), // Documentation Outline
    page.waitForNavigation(),
  ]);
})();
```



Navigation APIs



👉 `page.goto();`
👉 `page.waitForNavigation();`
👉 `page.goBack();`
👉 `page.goForward();`

Evaluation

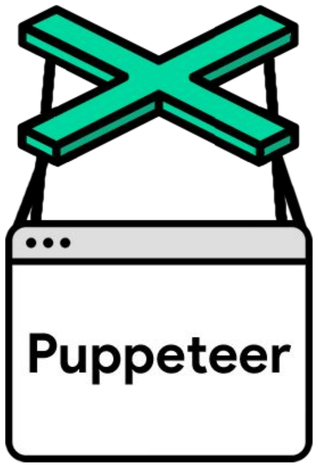
👉 evaluate JavaScript in website contexts

👉 use Web APIs to drive automation

HEISENBUG

Async!

Node.js Process



DevTools Protocol

Browser Process

HEISENBUG

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

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');
  const linkCount = await page.evaluate(() => {
    const links = document.querySelectorAll('a');
    return links.length;
  });
})();
```

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

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');
  const linkCount = await page.evaluate(() => {
    const links = document.querySelectorAll('a');
    return links.length;
  });
})();
```

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

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');
  const linkCount = await page.evaluate(() => {


  });
})();
```

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

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://joel.tools/merch');
  const linkCount = await page.evaluate(() => {
    browser; // undefined!
    page; // undefined!
  });
})();
```

Test Isolation

- 👉 Node.js **does not leak** into browser
- 👉 Website **doesn’t know** it’s been tested
- 👉 Websites **can’t ship test-specific** code

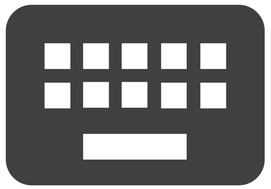
HEISENBUG

Takeaway

👉 Testing with Puppeteer is **foolproof**

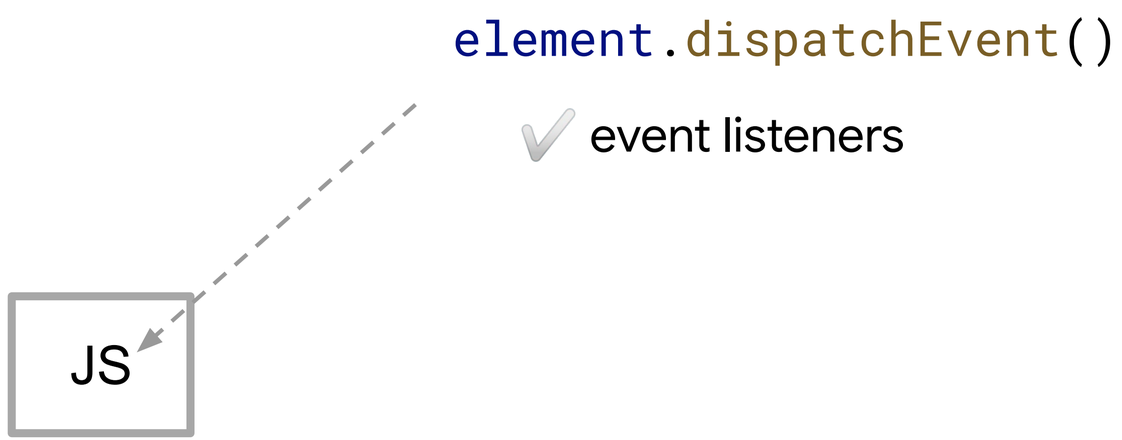
HEISENBUG

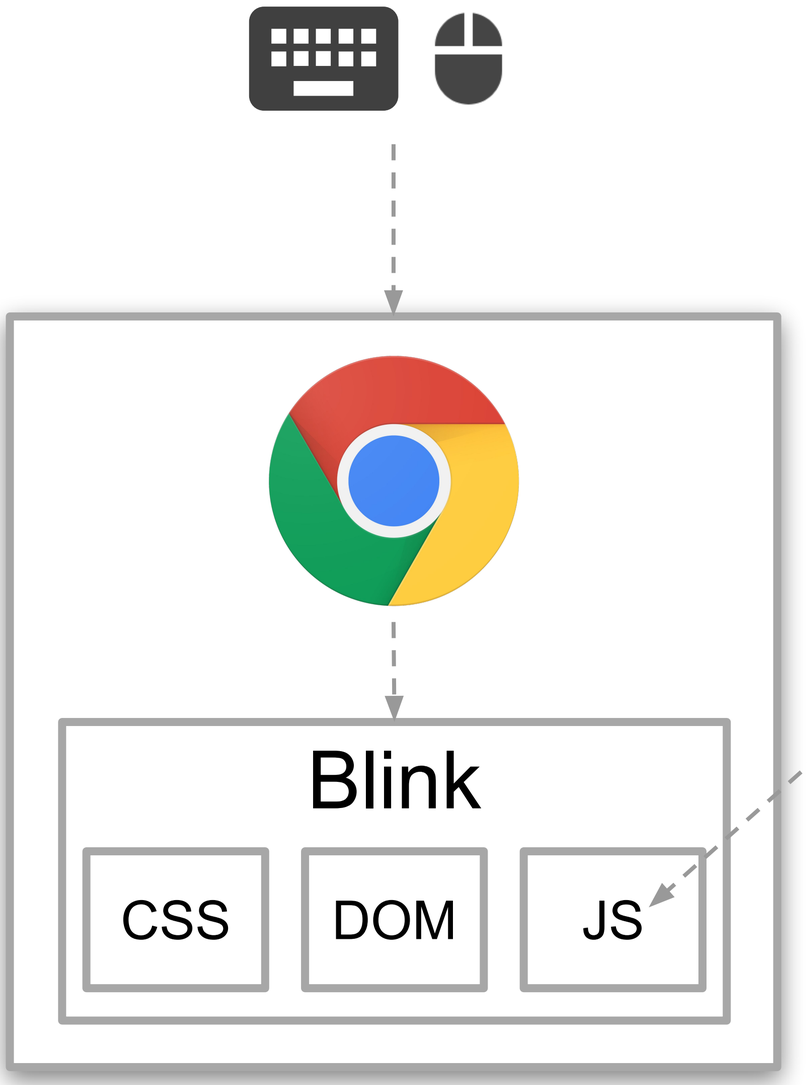
Input Injection



```
const mouseEvent = new MouseEvent('mousedown', {
  clientX: 50,
  clientY: 50,
});
document.body.dispatchEvent(mouseEvent);
```

```
const mouseEvent = new MouseEvent('mousedown', {
  clientX: 50,
  clientY: 50,
});
document.body.dispatchEvent(mouseEvent);
```

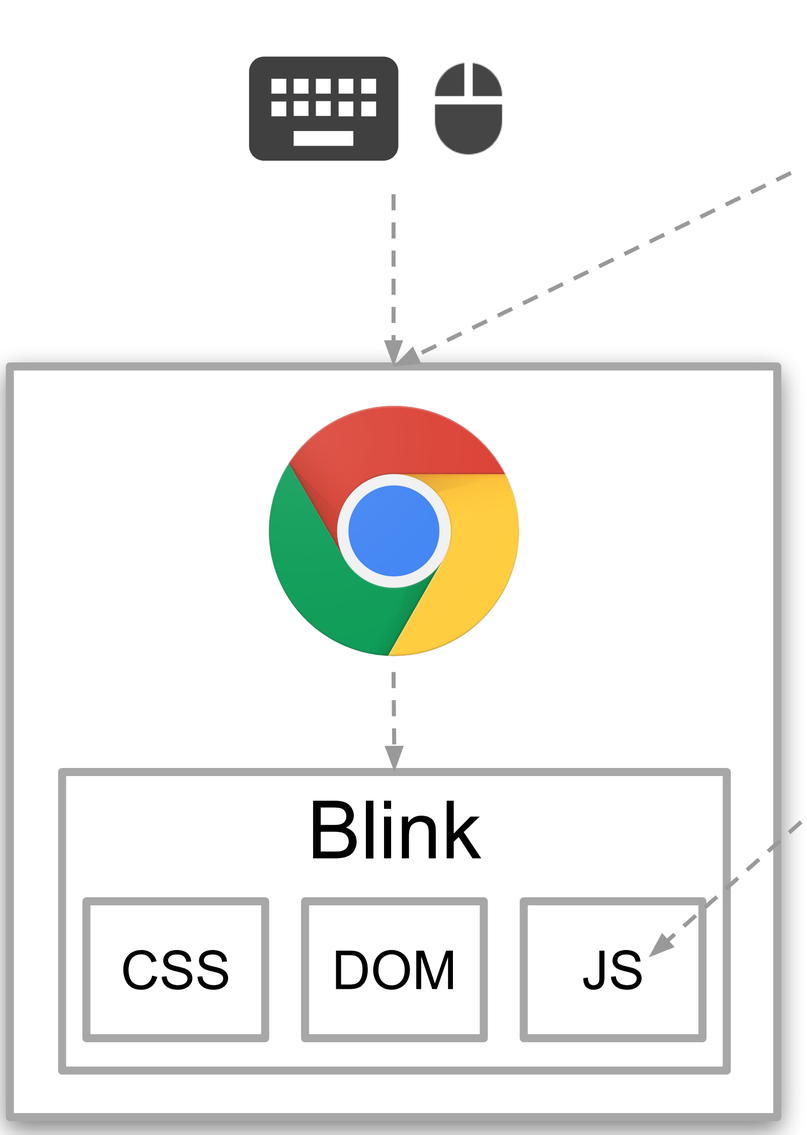


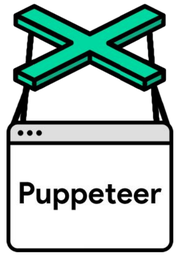



```
element.dispatchEvent()
```

- ✔️ event listeners
- 🚫 focus
- 🚫 selection
- 🚫 :hover
- 🚫 iframes
- 🚫 shadow DOM
- 🚫 insert text

HEISENBUG





- ✔ event listeners
- ✔ focus
- ✔ selection
- ✔ :hover
- ✔ iframes
- ✔ shadow DOM
- ✔ insert text

```
element.dispatchEvent()
```

- ✔ event listeners
- 🚫 focus
- 🚫 selection
- 🚫 :hover
- 🚫 iframes
- 🚫 shadow DOM
- 🚫 insert text

Input Injection APIs

👉 `page.keyboard.*` ⌨

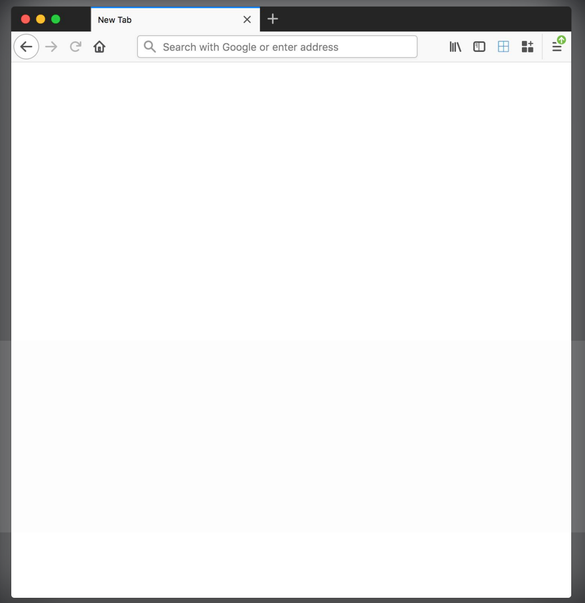
👉 `page.mouse.*` 🖱

👉 `page.type(selector, text);`

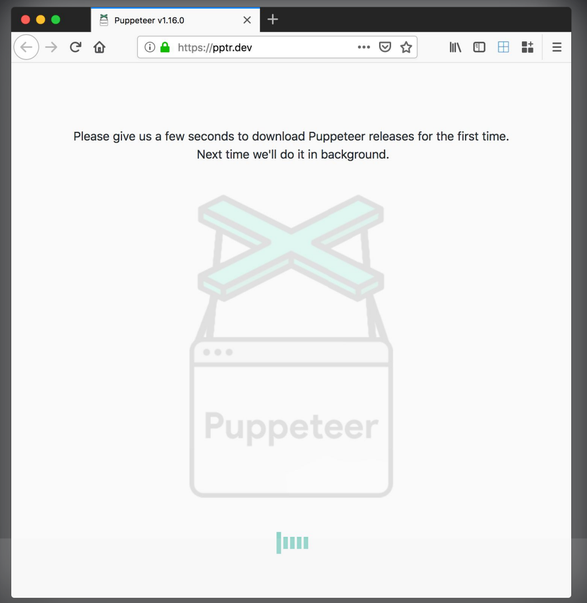
👉 `page.click(selector);`

```
const puppeteer = require('puppeteer-firefox');

(async () => {
  // Automate here




})();
```

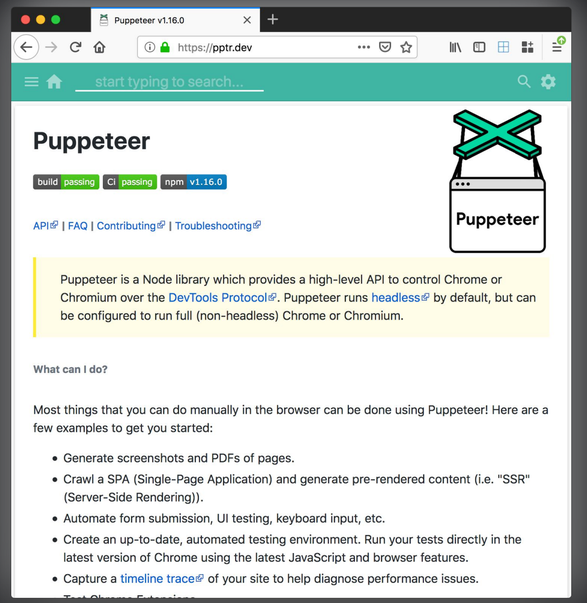
```
const puppeteer = require('puppeteer-firefox');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();

})();
```



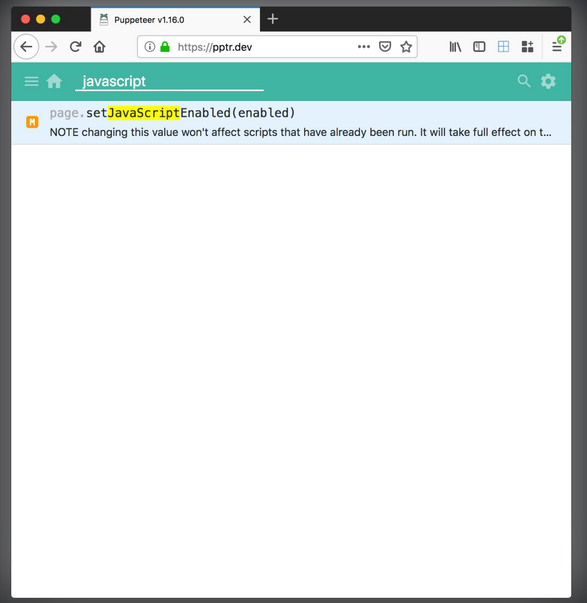
```
const puppeteer = require('puppeteer-firefox');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev');

})();
```



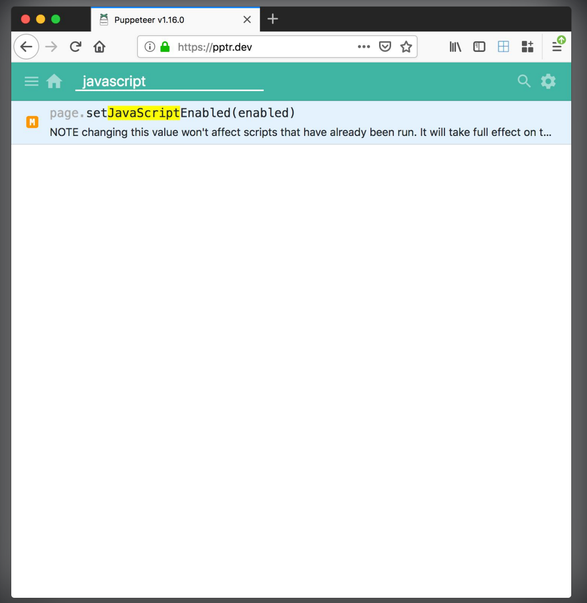
```
const puppeteer = require('puppeteer-firefox');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev');
  const input = await page.waitForSelector('[type=search]');

})();
```



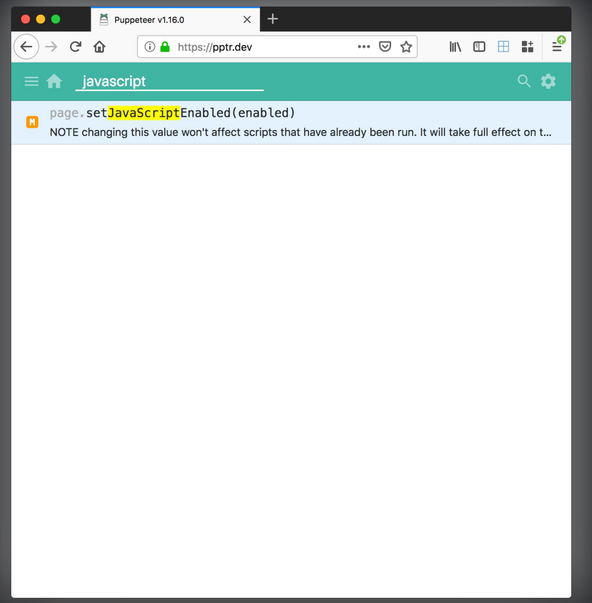
```
const puppeteer = require('puppeteer-firefox');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev');
  const input = await page.waitForSelector('[type=search]');
  await input.type('javascript');



})();
```



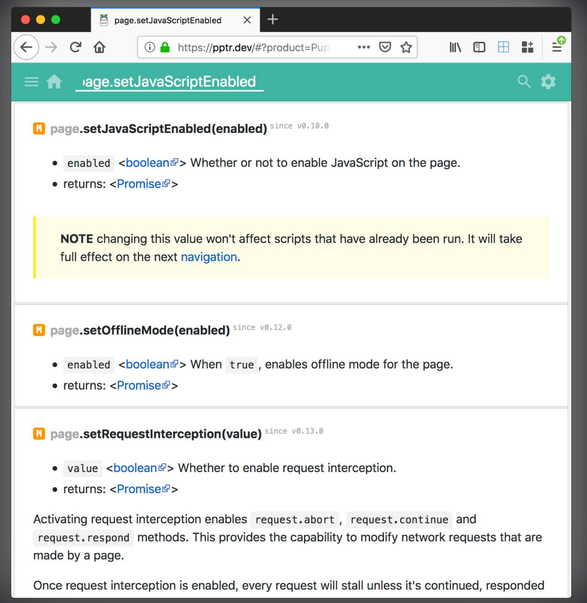
```
const puppeteer = require('puppeteer-firefox');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev');
  const input = await page.waitForSelector('[type=search]');
  await input.type('javascript');
  await Promise.all([



  ]);
})();
```

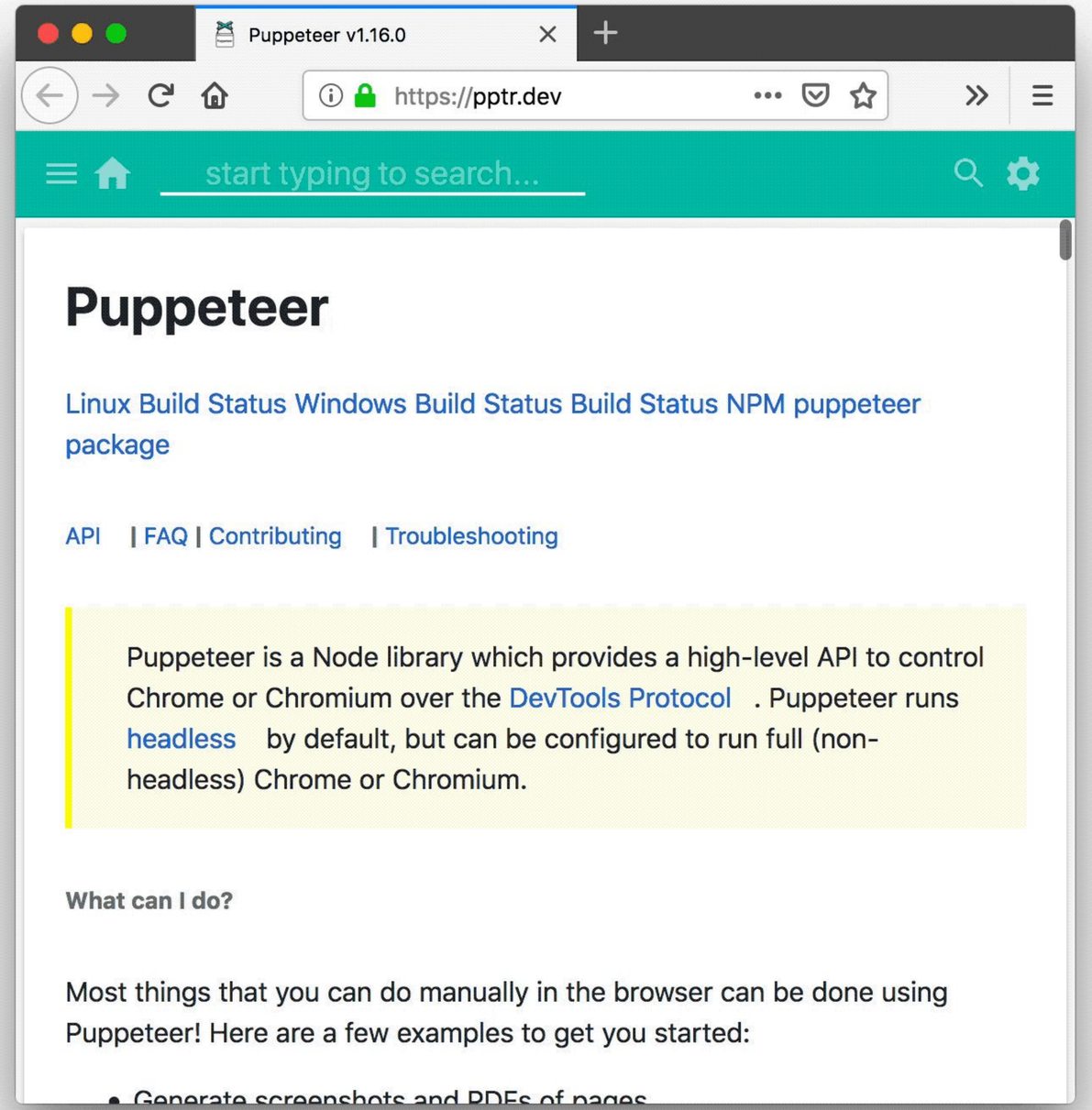


```
const puppeteer = require('puppeteer-firefox');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev');
  const input = await page.waitForSelector('[type=search]');
  await input.type('javascript');
  await Promise.all([
    page.keyboard.press('Enter'),

  ]);
})();
```



```
const puppeteer = require('puppeteer-firefox');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://pptr.dev');
  const input = await page.waitForSelector('[type=search]');
  await input.type('javascript');
  await Promise.all([
    page.keyboard.press('Enter'),
    page.waitForNavigation(),
  ]);
})();
```




Puppeteer v1.16.0
https://pptr.dev
start typing to search...
Puppeteer
Linux Build Status Windows Build Status Build Status NPM puppeteer package
API | FAQ | Contributing | Troubleshooting
Puppeteer is a Node library which provides a high-level API to control Chrome or Chromium over the DevTools Protocol . Puppeteer runs headless by default, but can be configured to run full (non-headless) Chrome or Chromium.
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:


Puppeteer

Summary

👉 All navigations are supported

👉 Tests are isolated from web content

👉 Real input

HEISENBUG

Testing Modern Web

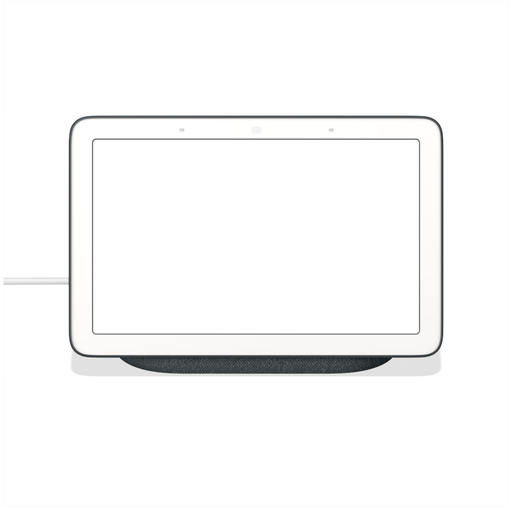
Modern Web

- 👉 WebApps
- 👉 Performance
- 👉 Accessibility

HEISENBUG

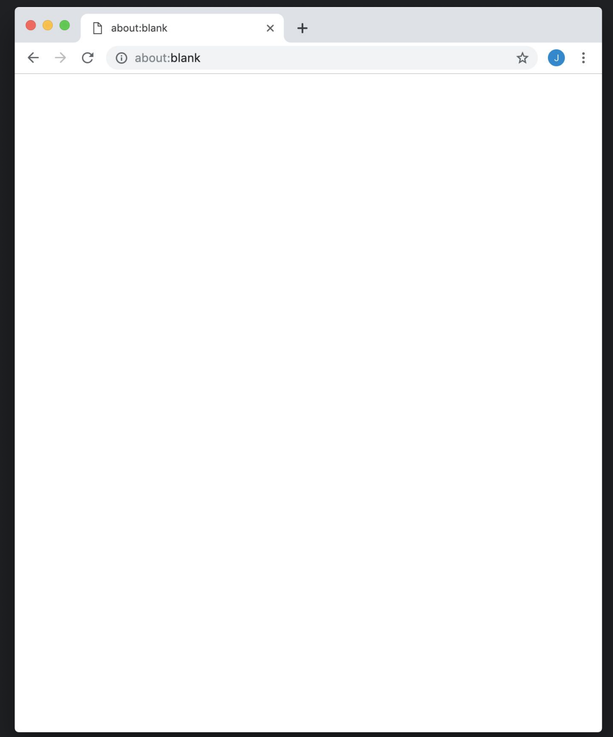
Mobile



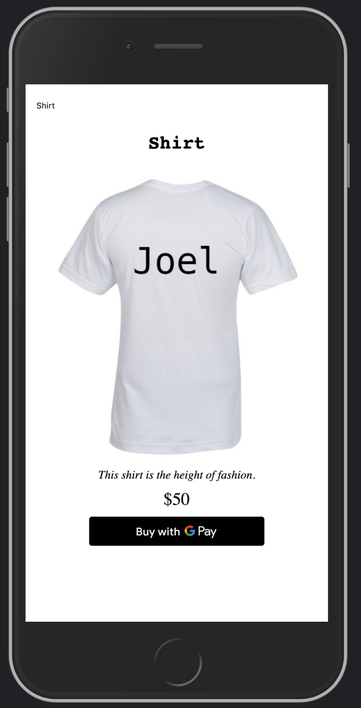
```
it('should work on mobile', async () => {
  // Test goes here




})
```

```
it('should work on mobile', async () => {
  const page = await context.newPage();



})
```

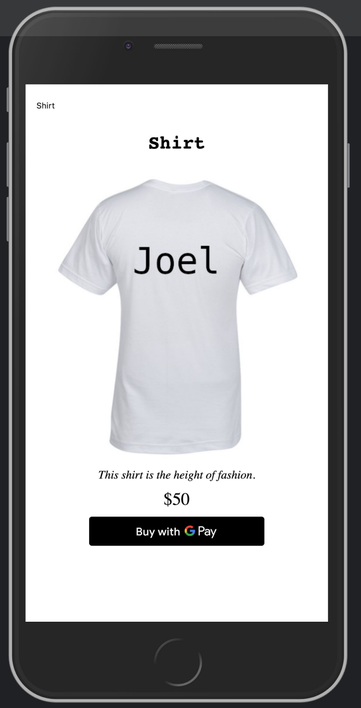


```
it('should work on mobile', async () => {
  const page = await context.newPage();
  await page.emulate(puppeteer.devices['iPhone 8']);



})
```

```
it('should work on mobile', async () => {
  const page = await context.newPage();
  await page.emulate(puppeteer.devices['iPhone 8']);
  await page.goto('https://joel.tools/merch/');


})
```



```
it('should work on mobile', async () => {
  const page = await context.newPage();
  await page.emulate(puppeteer.devices['iPhone 8']);
  await page.goto('https://joel.tools/merch/');
  const payButton = await page.waitForSelector('button.gpay-button');
  expect(payButton).toBeTruthy();
})
```



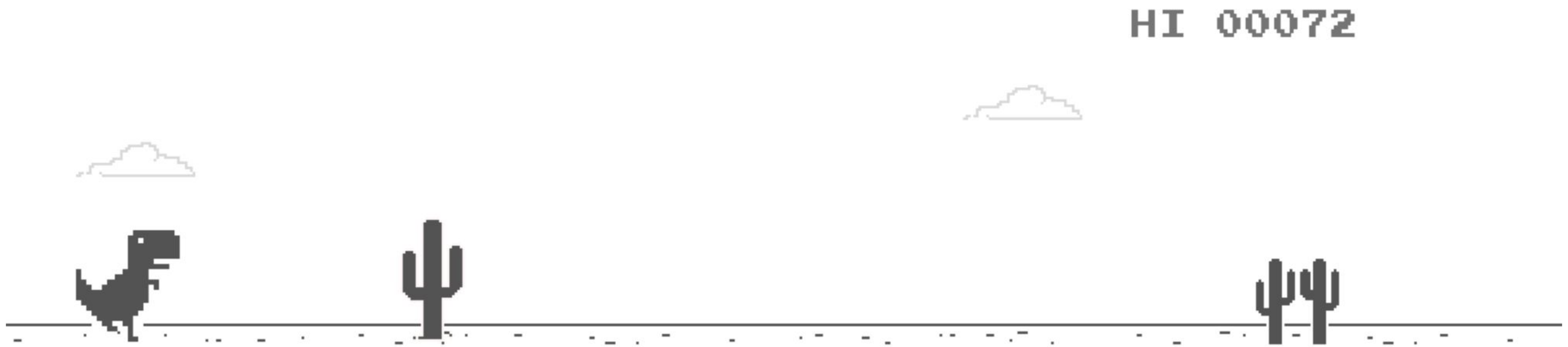
Device Emulation

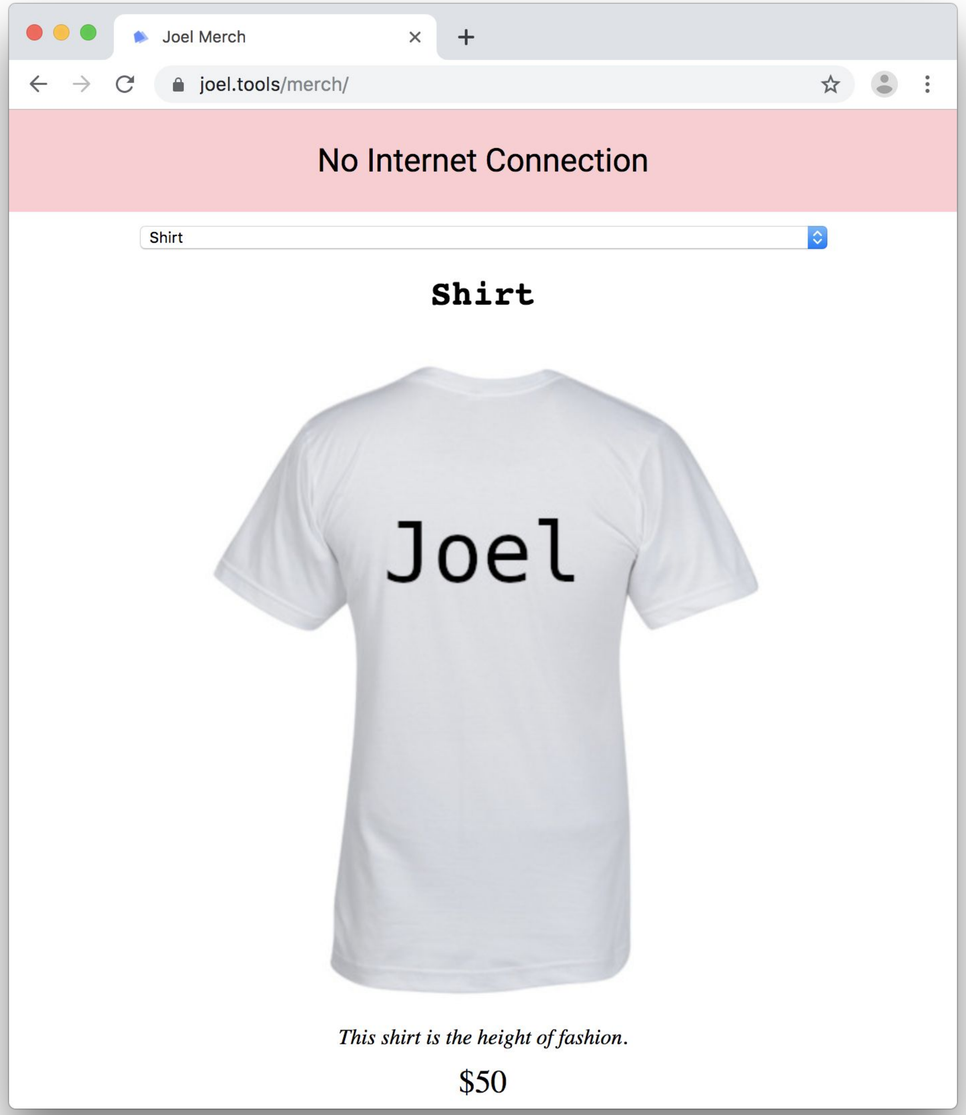
- 👉 User Agent
- 👉 Device Pixel Ratio
- 👉 Viewport size
- 👉 Touch Support

100+ devices

HEISENBUG

Offline Emulation

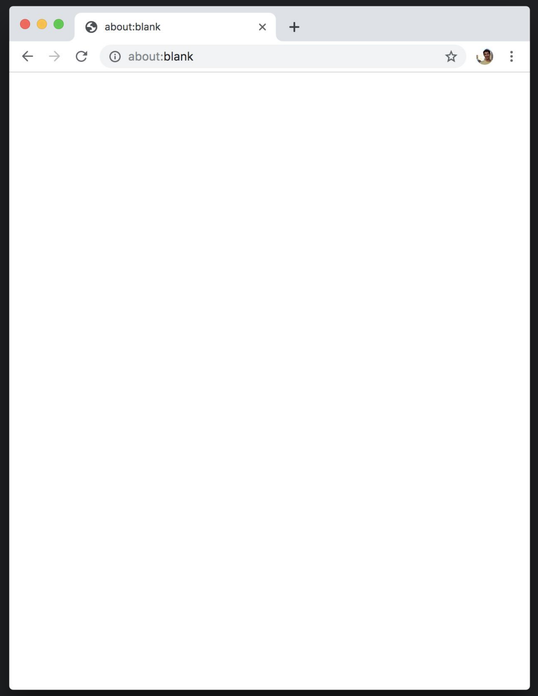



Joel Merch
joel.tools/merch/
No Internet Connection
Shirt
Shirt
Joel
This shirt is the height of fashion.
$50


Puppeteer

```
it('should notify user when goes offline', async () => {
  // Test goes here



});
```

```
it('should notify user when goes offline', async () => {
  const page = await context.newPage();


});
```



```
it('should notify user when goes offline', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch');



});
```

```
it('should notify user when goes offline', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch');
  await page.setOfflineMode(true);


});
```

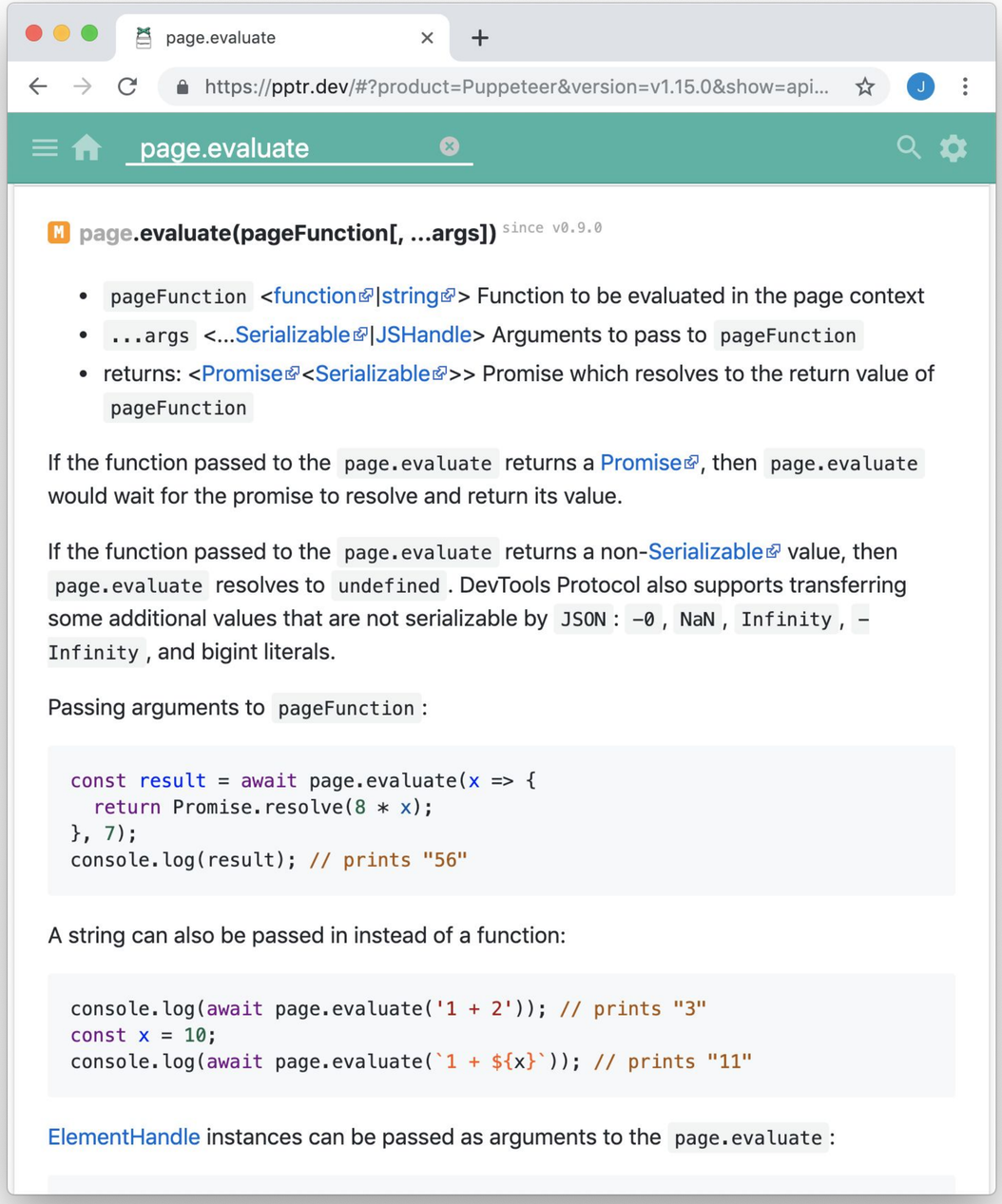
```
it('should notify user when goes offline', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch');
  await page.setOfflineMode(true);
  const msg = await page.waitForSelector('.offline-alert');
  expect(msg).toBeTruthy();
});
```

Service Workers

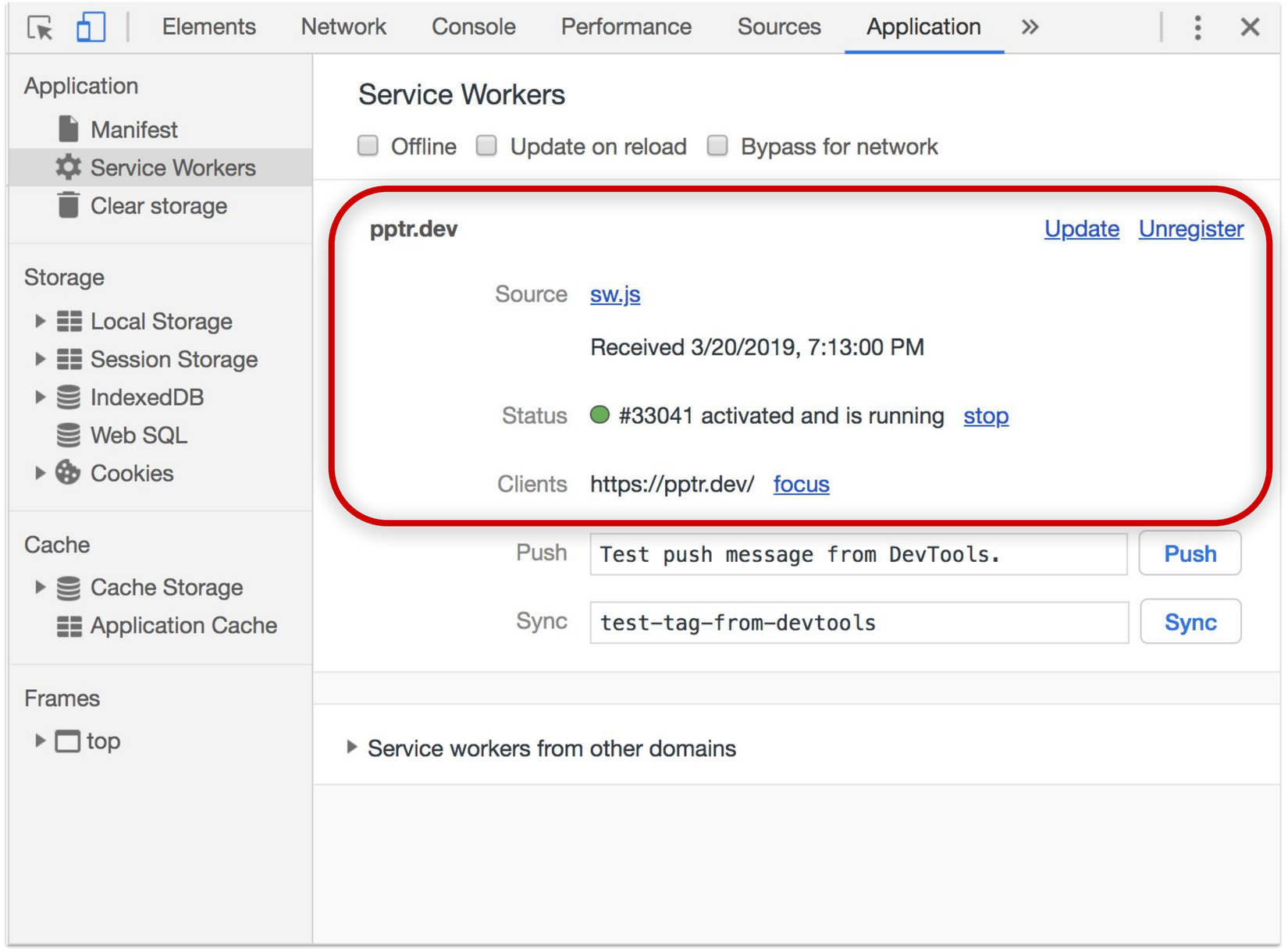
HEISENBUG

pptr.dev

Workbox



HEISENBUG

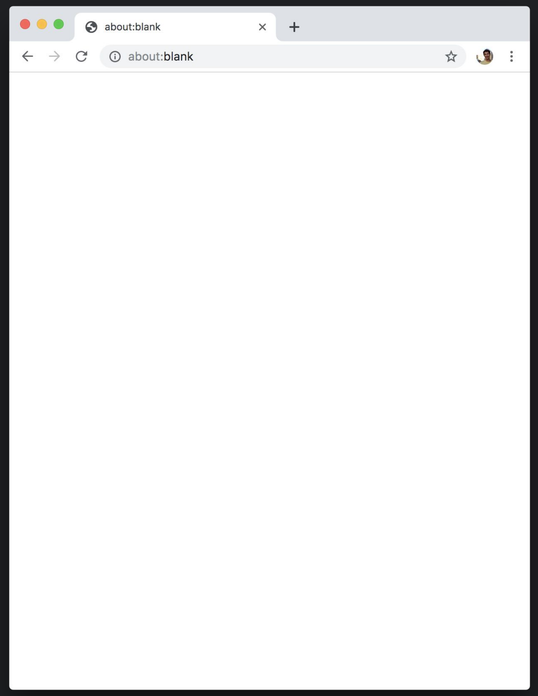

Elements
Network
Console
Performance
Sources
Application
Application
Manifest
Service Workers
Clear storage
Storage
Local Storage
Session Storage
IndexedDB
Web SQL
Cookies
Cache
Cache Storage
Application Cache
Frames
top
Service Workers
Offline
Update on reload
Bypass for network
pptr.dev
Update
Unregister
Source
sw.js
Received 3/20/2019, 7:13:00 PM
Status
#33041 activated and is running
stop
Clients
https://pptr.dev/
focus
Push
Test push message from DevTools.
Push
Sync
test-tag-from-devtools
Sync
Service workers from other domains


Puppeteer

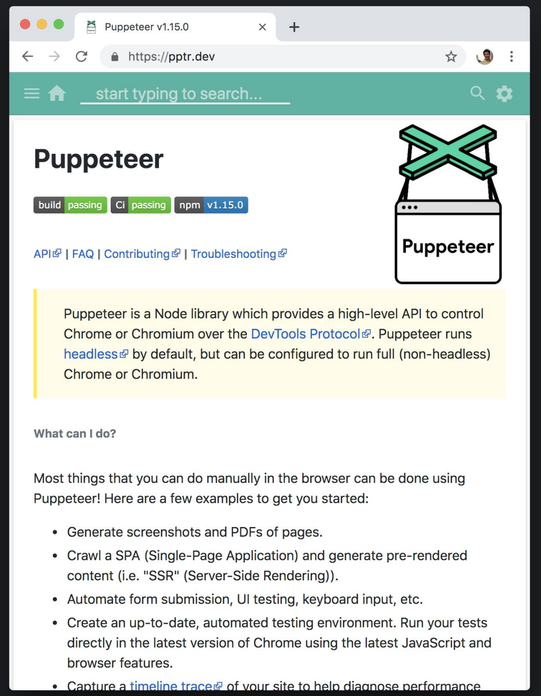
HEISENBUG

```
it('should register a service worker', async () => {
  // Test goes here




})
```

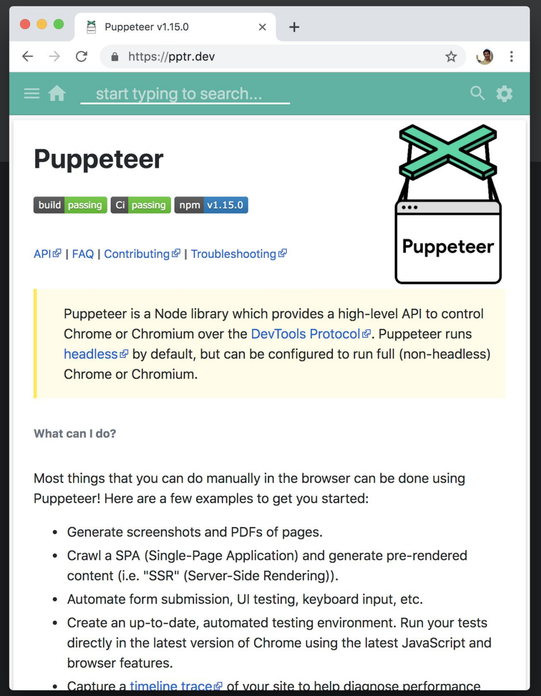
```
it('should register a service worker', async () => {
  const page = await context.newPage();



})
```



```
it('should register a service worker', async () => {
  const page = await context.newPage();
  await page.goto('https://pptr.dev');


})
```



```
it('should register a service worker', async () => {
  const page = await context.newPage();
  await page.goto('https://pptr.dev');
  const swTarget = await context.waitForTarget(target => {
    return target.type() === 'service_worker';
  });
})
```



```
...
const swTarget = await context.waitForTarget(target => {
  return target.type() === 'service_worker';
});
```

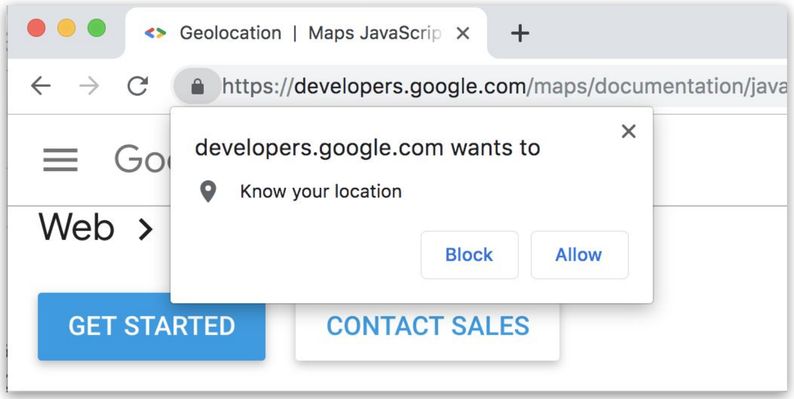
```
...
const swTarget = await context.waitForTarget(target => {
  return target.type() === 'service_worker';
});
const serviceWorker = await swTarget.worker();
```

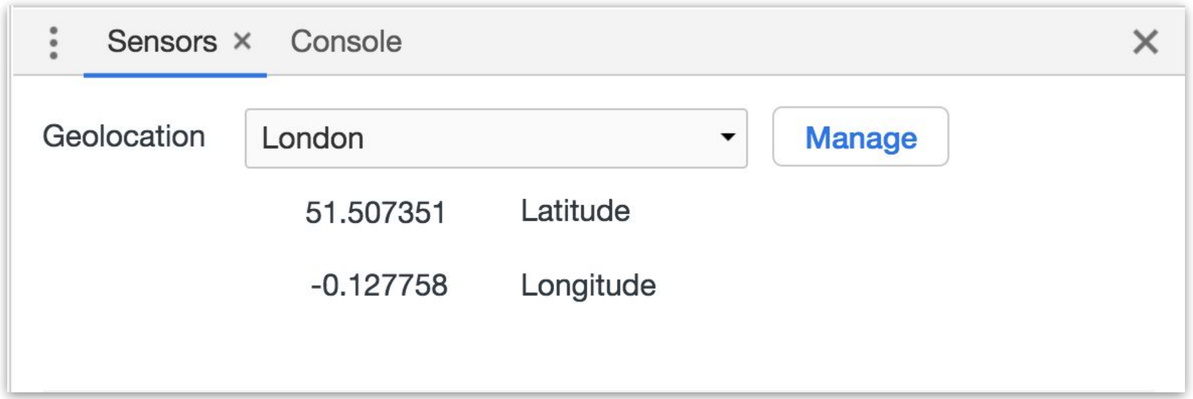
```
...
const swTarget = await context.waitForTarget(target => {
  return target.type() === 'service_worker';
});
const serviceWorker = await swTarget.worker();
const isCached = await serviceWorker.evaluate(async () => {
  return !!(await caches.match('https://pptr.dev/logo.png'));
});
expect(isCached).toBe(true);
```

Geolocation

HEISENBUG

Geolocation: London





HEISENBUG

```
it('should have prices in pounds in London', async () => {
  // Test goes here



})
```

```
it('should have prices in pounds in London', async () => {
  await context.overridePermissions('https://joel.tools', [
    'geolocation',
  ]);









})
```

```
it('should have prices in pounds in London', async () => {
  await context.overridePermissions('https://joel.tools', [
    'geolocation',
  ]);
  const page = await context.newPage();




})
```

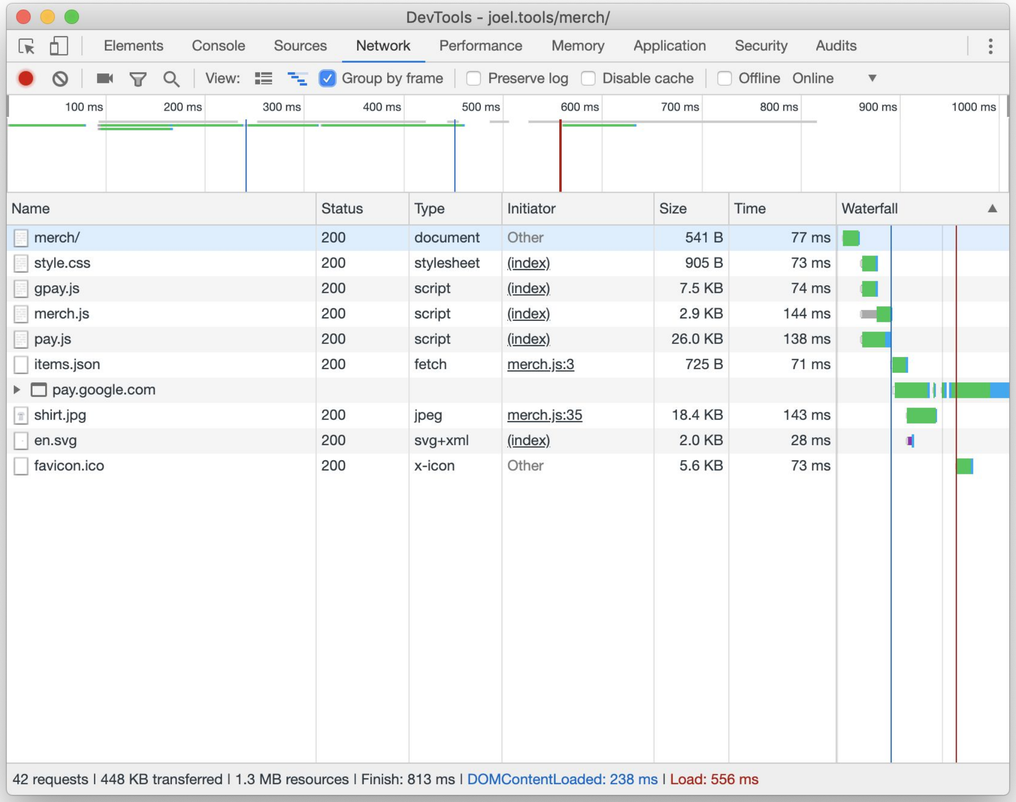
```
it('should have prices in pounds in London', async () => {
  await context.overridePermissions('https://joel.tools', [
    'geolocation',
  ]);
  const page = await context.newPage();
  await page.setGeolocation({
    latitude: 51.50,
    longitude: -0.12,
  });



})
```

```
it('should have prices in pounds in London', async () => {
  await context.overridePermissions('https://joel.tools', [
    'geolocation',
  ]);
  const page = await context.newPage();
  await page.setGeolocation({
    latitude: 51.50,
    longitude: -0.12,
  });
  await page.goto('https://joel.tools/merch');


})
```

```
it('should have prices in pounds in London', async () => {
  await context.overridePermissions('https://joel.tools', [
    'geolocation',
  ]);
  const page = await context.newPage();
  await page.setGeolocation({
    latitude: 51.50,
    longitude: -0.12,
  });
  await page.goto('https://joel.tools/merch');
  const price = await page.$eval('.price', div => div.textContent);
  expect(price).toBe('£40');
})
```

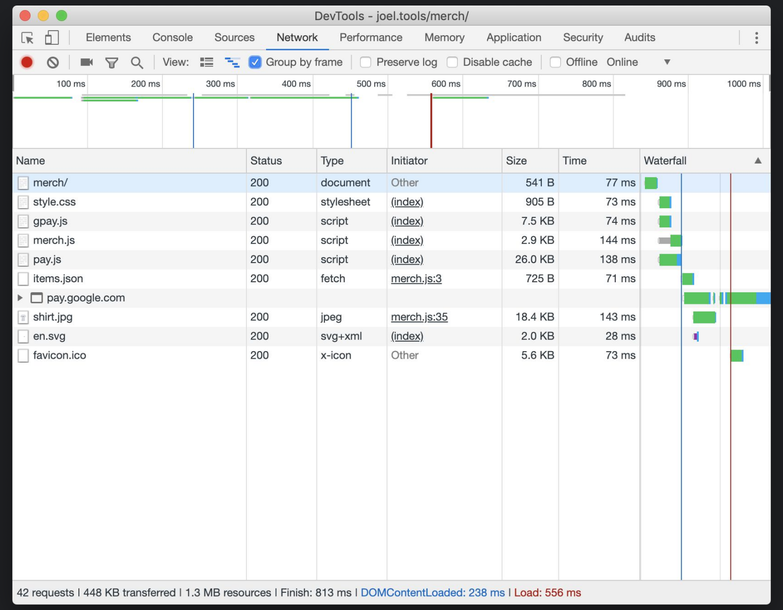
£40

Network Monitoring



HEISENBUG

```
page.on('request', request => {
  console.log(request.method(), request.url());
});
page.on('response', response => {
  console.log(response.status(), response.url());
});
```

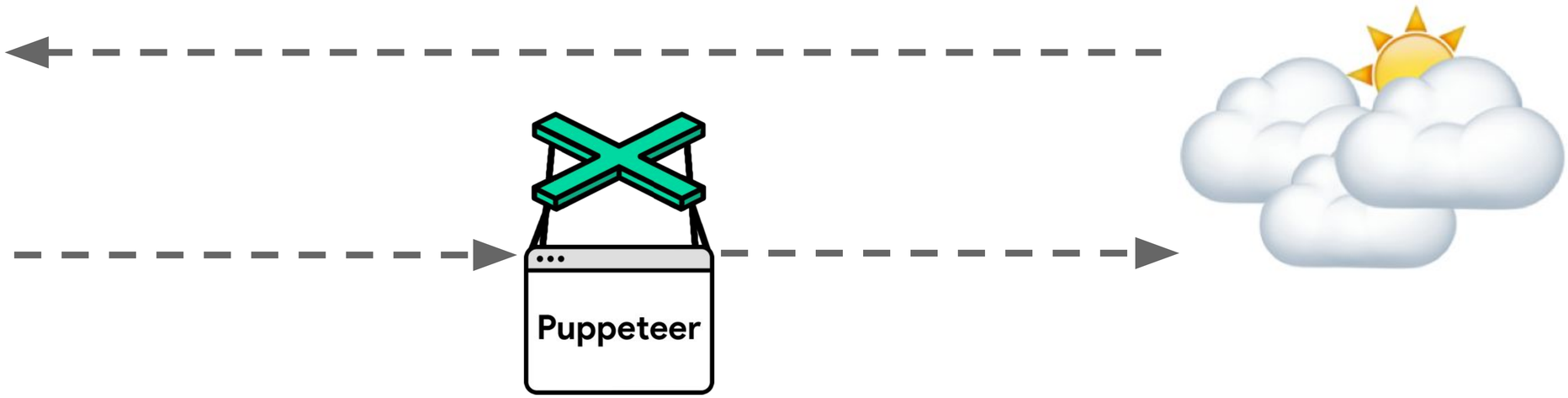


Network Monitoring

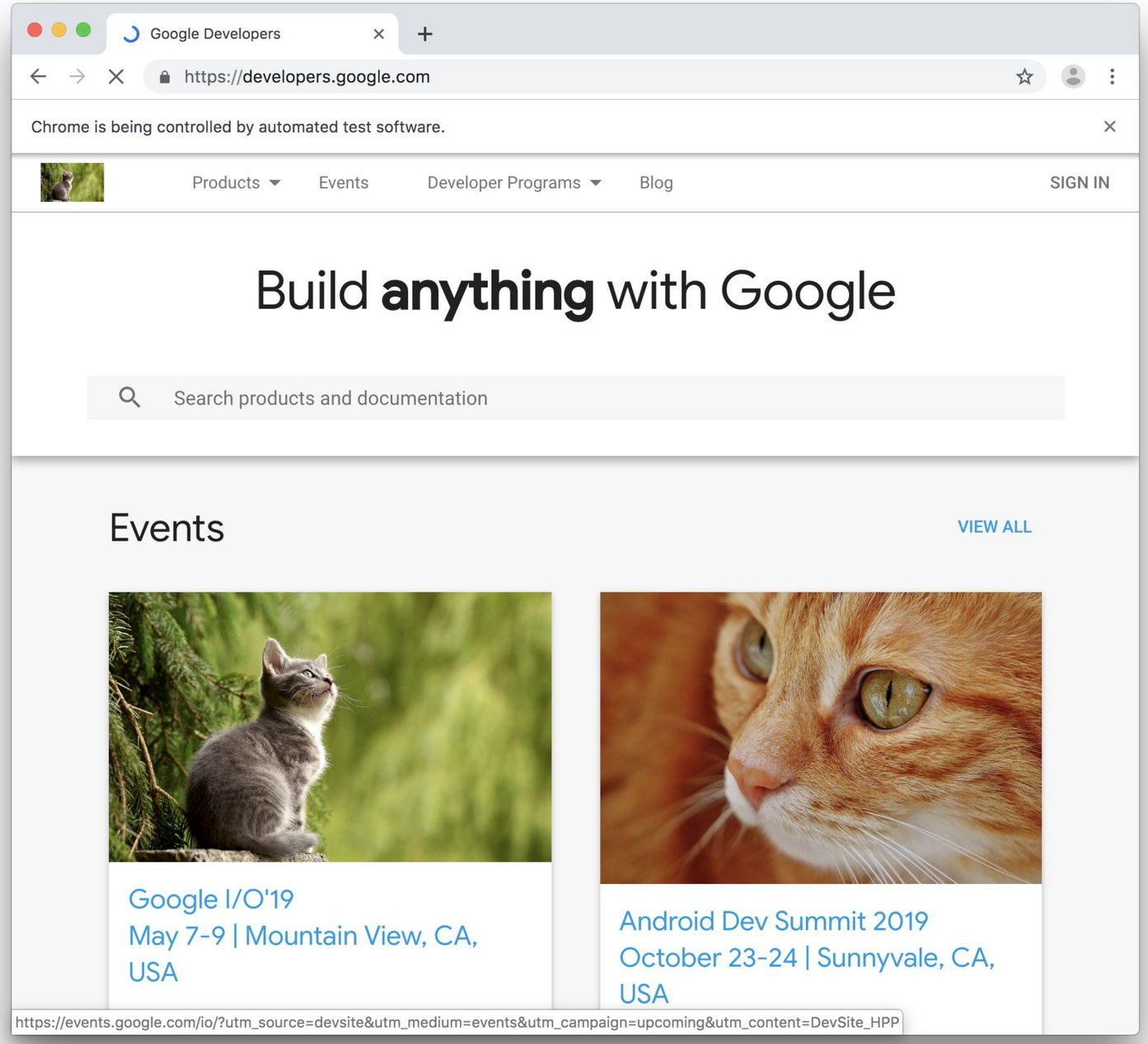
- 👉 Headers
- 👉 POST Data
- 👉 Content
- 👉 From cache / From Service Worker

HEISENBUG

Request Interception

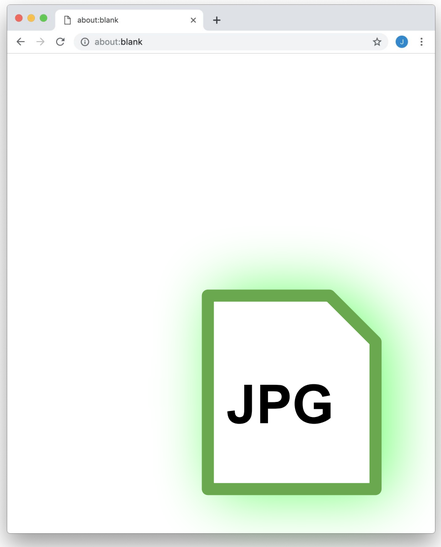


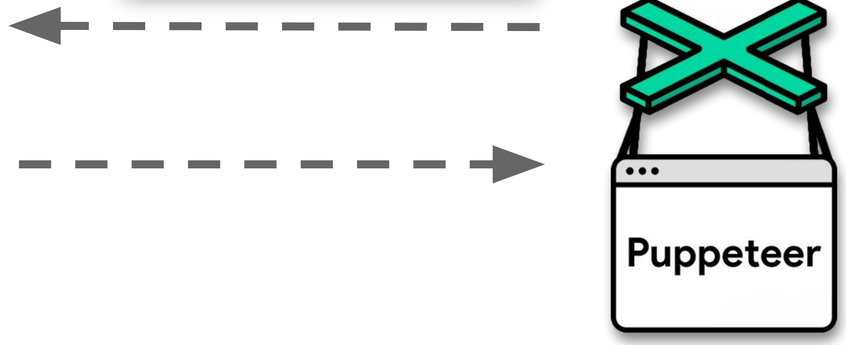
HEISENBUG


Google Developers
https://developers.google.com
Chrome is being controlled by automated test software.
Products
Events
Developer Programs
Blog
SIGN IN
Build anything with Google
Search products and documentation
Events
VIEW ALL
Google I/O'19
May 7-9 | Mountain View, CA, USA
Android Dev Summit 2019
October 23-24 | Sunnyvale, CA, USA
https://events.google.com/io/?utm_source=devsite&utm_medium=events&utm_campaign=upcoming&utm_content=DevSite_HPP

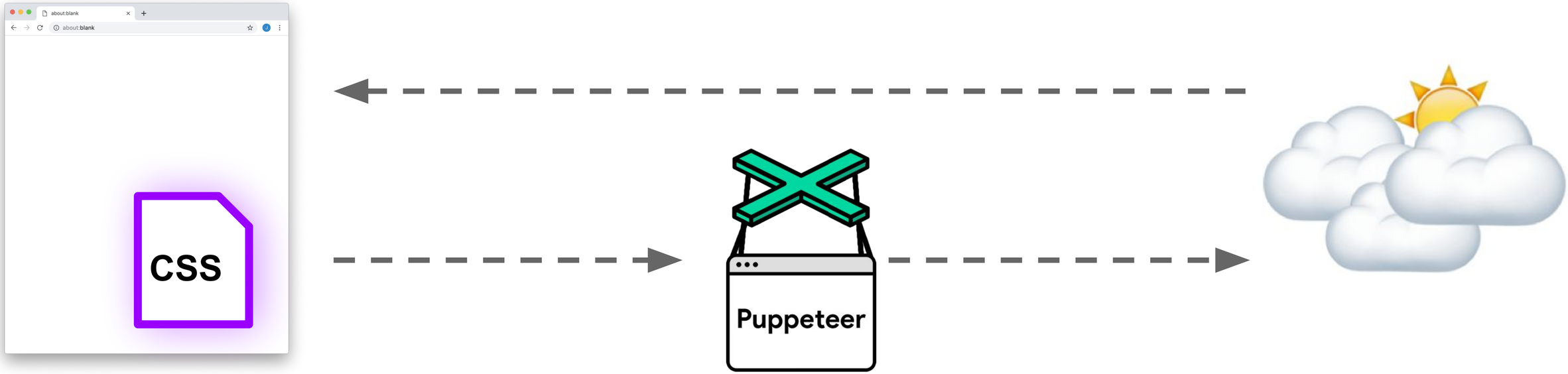

Puppeteer

HEISENBUG


about:blank
about:blank
JPG


Puppeteer


Puppeteer


about:blank
about:blank
CSS
Puppeteer


Puppeteer

```
await page.setRequestInterception(true);
page.on('request', request => {
  if (request.resourceType() === 'image')
    request.respond({ body: randomCatImage() });
  else
    request.continue();
});
```

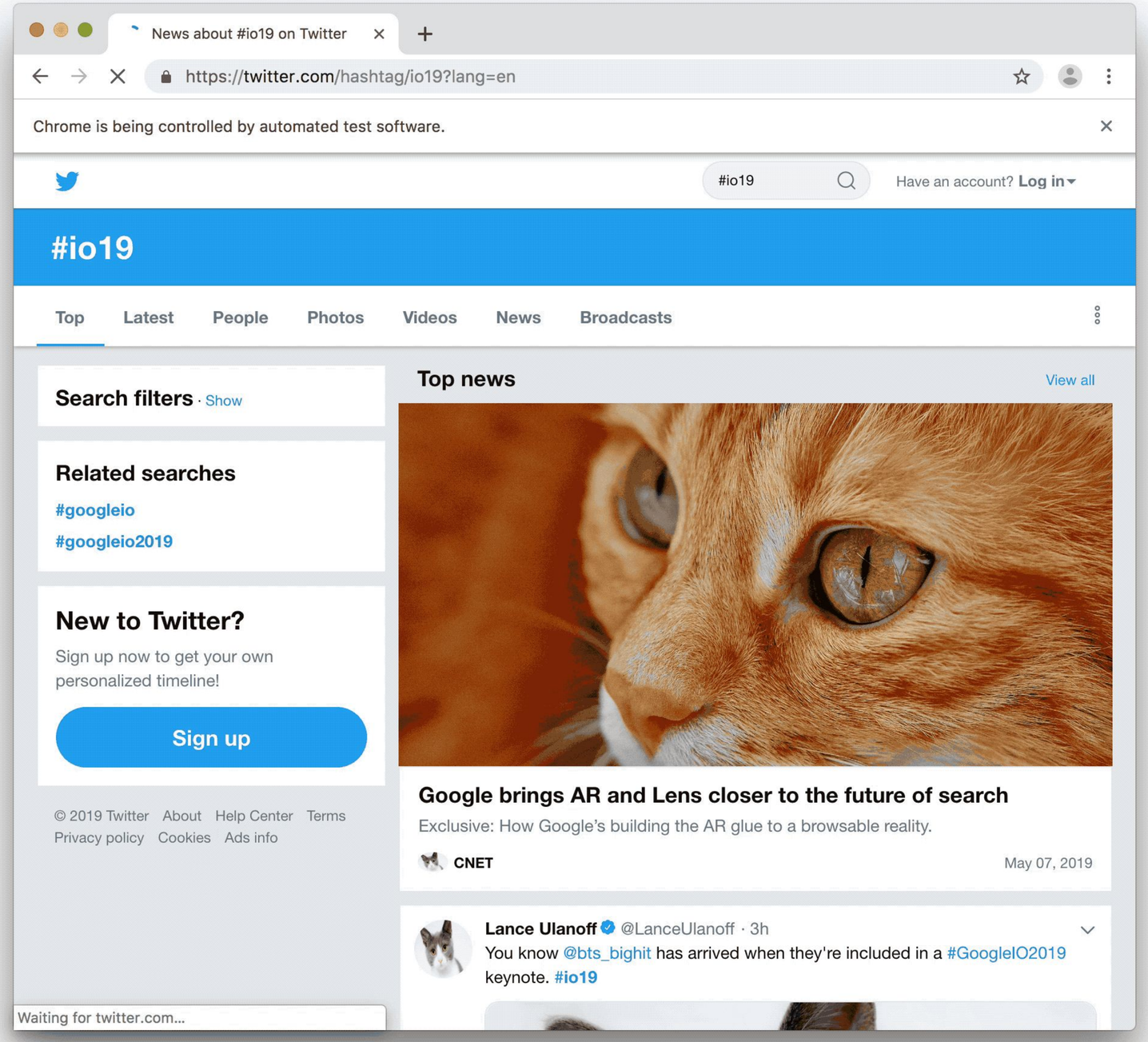
```
await page.setRequestInterception(true);
page.on('request', request => {
  if (request.resourceType() === 'image')
    request.respond({ body: randomCatImage() });
  else
    request.continue();
});
```

```
await page.setRequestInterception(true);
page.on('request', request => {
  if (request.resourceType() === 'image')
    request.respond({ body: randomCatImage() });
  else
    request.continue();
});
```

```
await page.setRequestInterception(true);
page.on('request', request => {
  if (request.resourceType() === 'image')
    request.respond({ body: randomCatImage() });
  else
    request.continue();
});
```

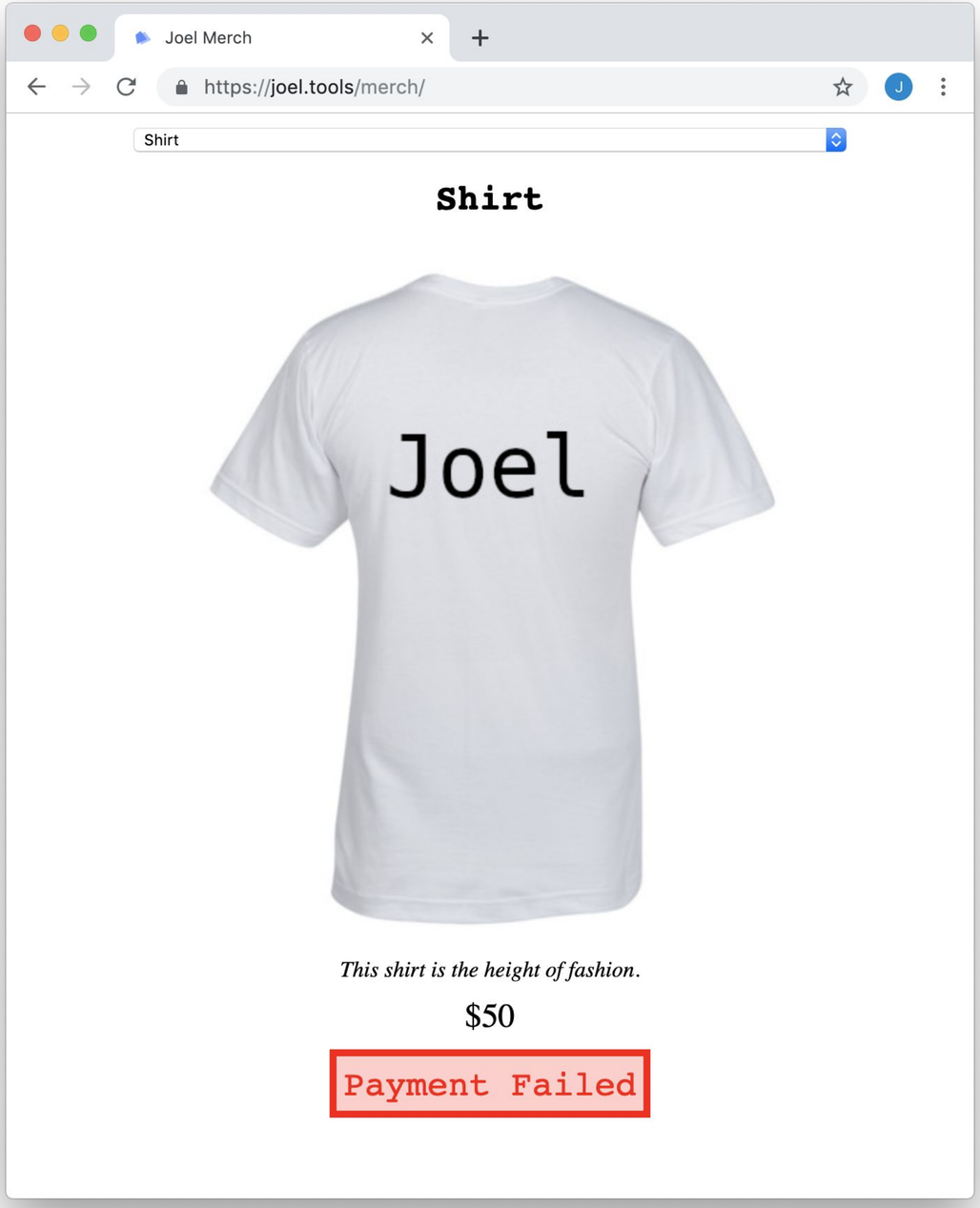
```
await page.setRequestInterception(true);
page.on('request', request => {
  if (request.resourceType() === 'image')
    request.respond({ body: randomCatImage() });
  else
    request.continue();
});
```

```
await page.setRequestInterception(true);
page.on('request', request => {
  if (request.resourceType() === 'image')
    request.respond({ body: randomCatImage() });
  else
    request.continue();
});
```


News about #io19 on Twitter
https://twitter.com/hashtag/io19?lang=en
Chrome is being controlled by automated test software.
#io19
Have an account? Log in
#io19
Top
Latest
People
Photos
Videos
News
Broadcasts
Search filters · Show
Related searches
#googleio
#googleio2019
New to Twitter?
Sign up now to get your own personalized timeline!
Sign up
© 2019 Twitter About Help Center Terms
Privacy policy Cookies Ads info
Top news
View all
Google brings AR and Lens closer to the future of search
Exclusive: How Google's building the AR glue to a browsable reality.
CNET
May 07, 2019
Lance Ulanoff @LanceUlanoff · 3h
You know @bts_bighit has arrived when they're included in a #GoogleIO2019 keynote. #io19
Waiting for twitter.com...


Puppeteer

HEISENBUG

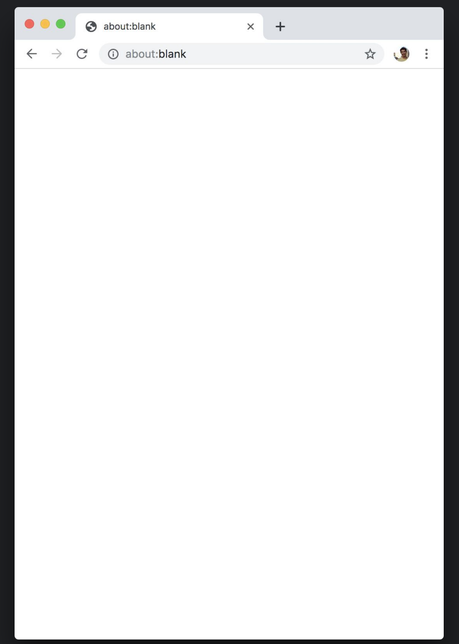

Joel Merch
https://joel.tools/merch/
Shirt
Shirt
Joel
This shirt is the height of fashion.
$50
Payment Failed


Puppeteer

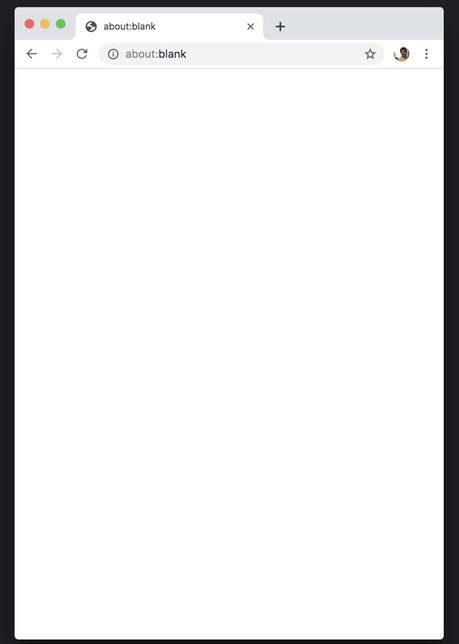
HEISENBUG

```
it('should notify when payment fails', async () => {
  // Test goes here




})
```

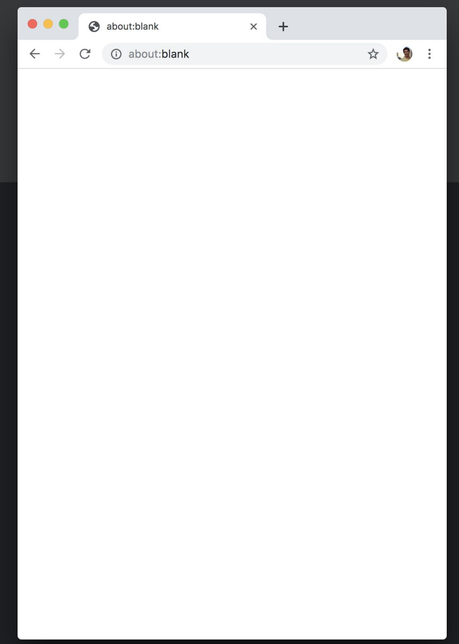
```
it('should notify when payment fails', async () => {
  const page = await context.newPage();


})
```



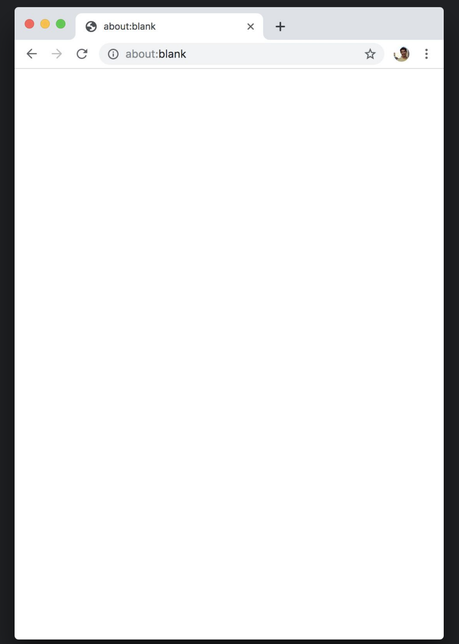
```
it('should notify when payment fails', async () => {
  const page = await context.newPage();
  await page.setRequestInterception(true);


})
```



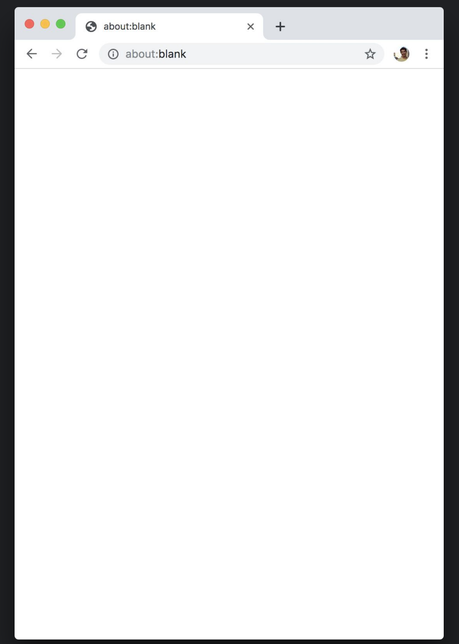
```
it('should notify when payment fails', async () => {
  const page = await context.newPage();
  await page.setRequestInterception(true);
  page.on('request', request => {




  });




})
```



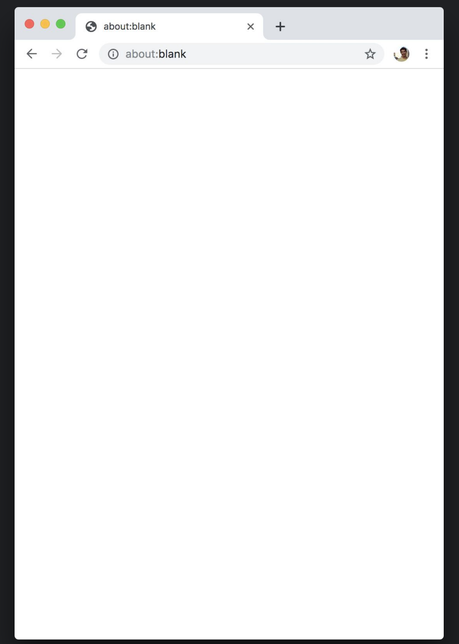
```
it('should notify when payment fails', async () => {
  const page = await context.newPage();
  await page.setRequestInterception(true);
  page.on('request', request => {
    if (request.url() === 'https://joel.tools/pay')




  });




})
```



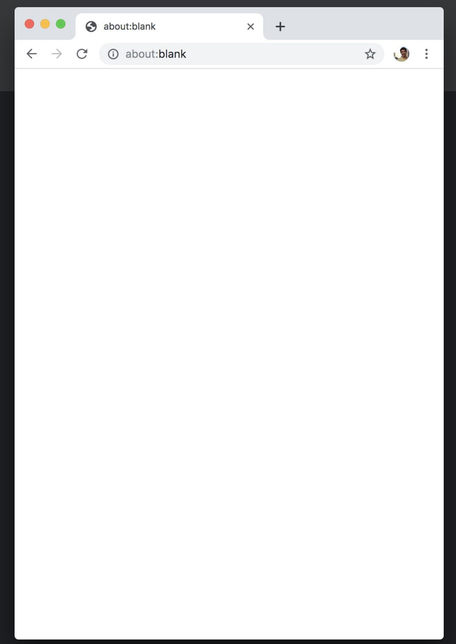
```
it('should notify when payment fails', async () => {
  const page = await context.newPage();
  await page.setRequestInterception(true);
  page.on('request', request => {
    if (request.url() === 'https://joel.tools/pay')
      request.abort();


  });




})
```



```
it('should notify when payment fails', async () => {
  const page = await context.newPage();
  await page.setRequestInterception(true);
  page.on('request', request => {
    if (request.url() === 'https://joel.tools/pay')
      request.abort();
    else


  });




})
```



```
it('should notify when payment fails', async () => {
  const page = await context.newPage();
  await page.setRequestInterception(true);
  page.on('request', request => {
    if (request.url() === 'https://joel.tools/pay')
      request.abort();
    else
      request.continue();
  });



})
```



```
it('should notify when payment fails', async () => {
  const page = await context.newPage();
  await page.setRequestInterception(true);
  page.on('request', request => {
    if (request.url() === 'https://joel.tools/pay')
      request.abort();
    else
      request.continue();
  });

  await page.goto('https://joel.tools/merch');


})
```

```
it('should notify when payment fails', async () => {
  const page = await context.newPage();
  await page.setRequestInterception(true);
  page.on('request', request => {
    if (request.url() === 'https://joel.tools/pay')
      request.abort();
    else
      request.continue();
  });

  await page.goto('https://joel.tools/merch');
  await page.click('button.gpay-button');

})
```

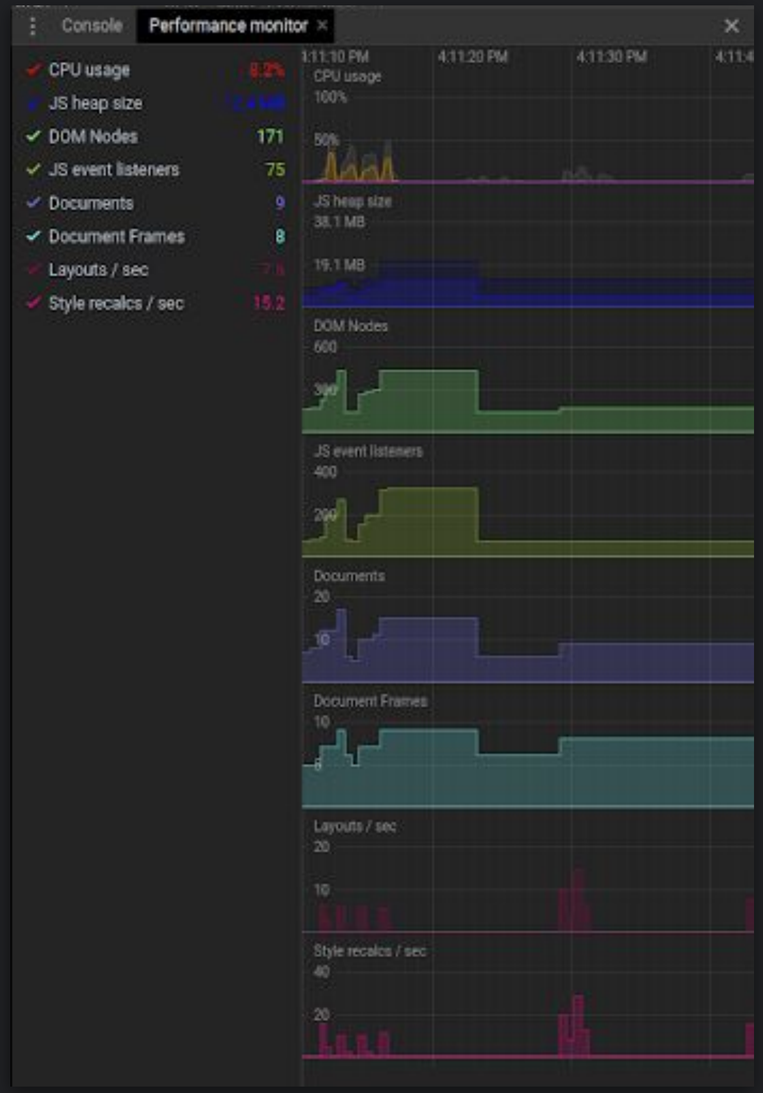
```
it('should notify when payment fails', async () => {
  const page = await context.newPage();
  await page.setRequestInterception(true);
  page.on('request', request => {
    if (request.url() === 'https://joel.tools/pay')
      request.abort();
    else
      request.continue();
  });

  await page.goto('https://joel.tools/merch');
  await page.click('button.gpay-button');
  await page.waitForSelector('.payment-failed');
})
```

Performance Testing

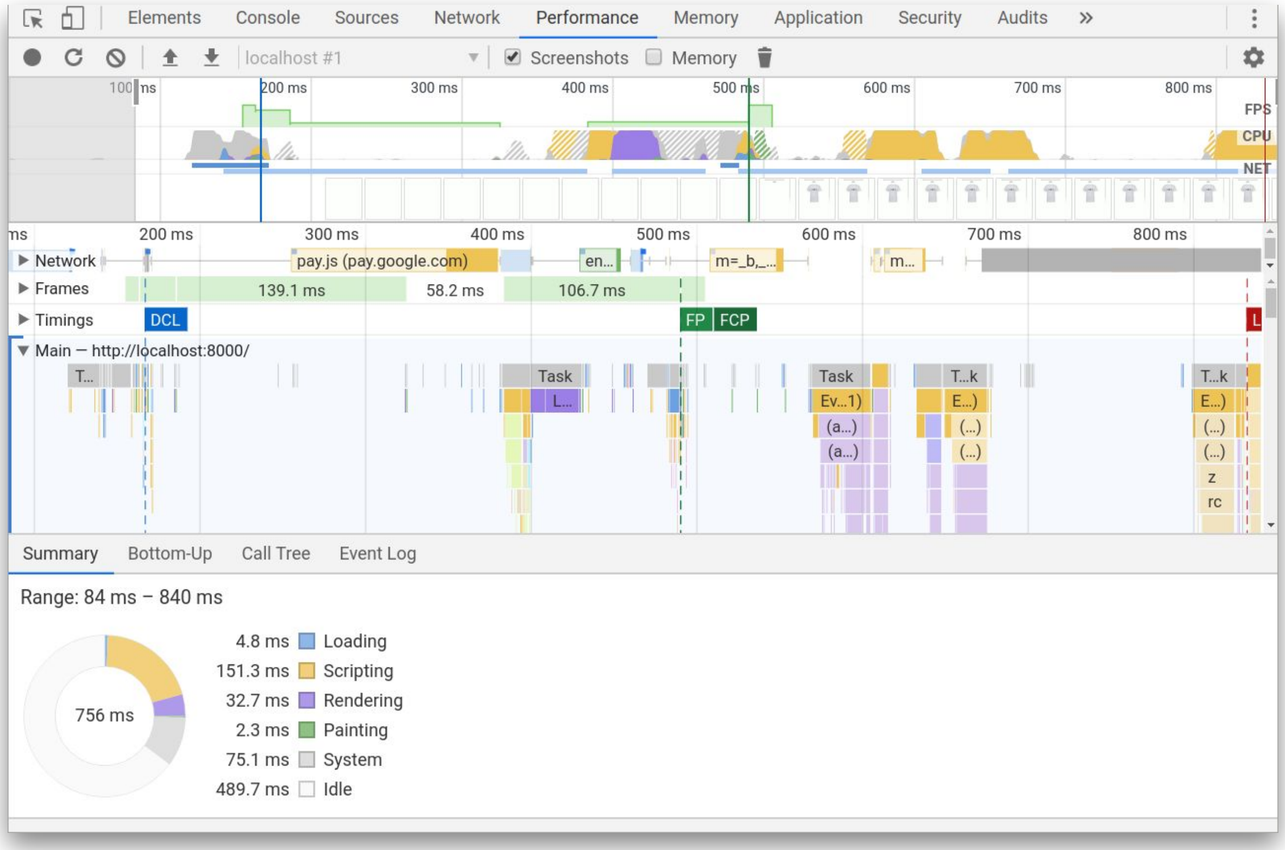
👉 Metrics
👉 Chrome Tracing
👉 Code Coverage

HEISENBUG

```
const metrics = await page.metrics();
console.log(metrics);

{ Timestamp: 3180099.406978,
  Documents: 7,
  Frames: 3,
  JSEventListeners: 121,
  Nodes: 156,
  LayoutCount: 6,
  RecalcStyleCount: 7,
  LayoutDuration: 0.021108,
  RecalcStyleDuration: 0.009698,
  ScriptDuration: 0.158487,
  TaskDuration: 0.401342,
  JSHeapUsedSize: 7107312,
  JSHeapTotalSize: 11689984 }
```



Chrome Tracing



Puppeteer

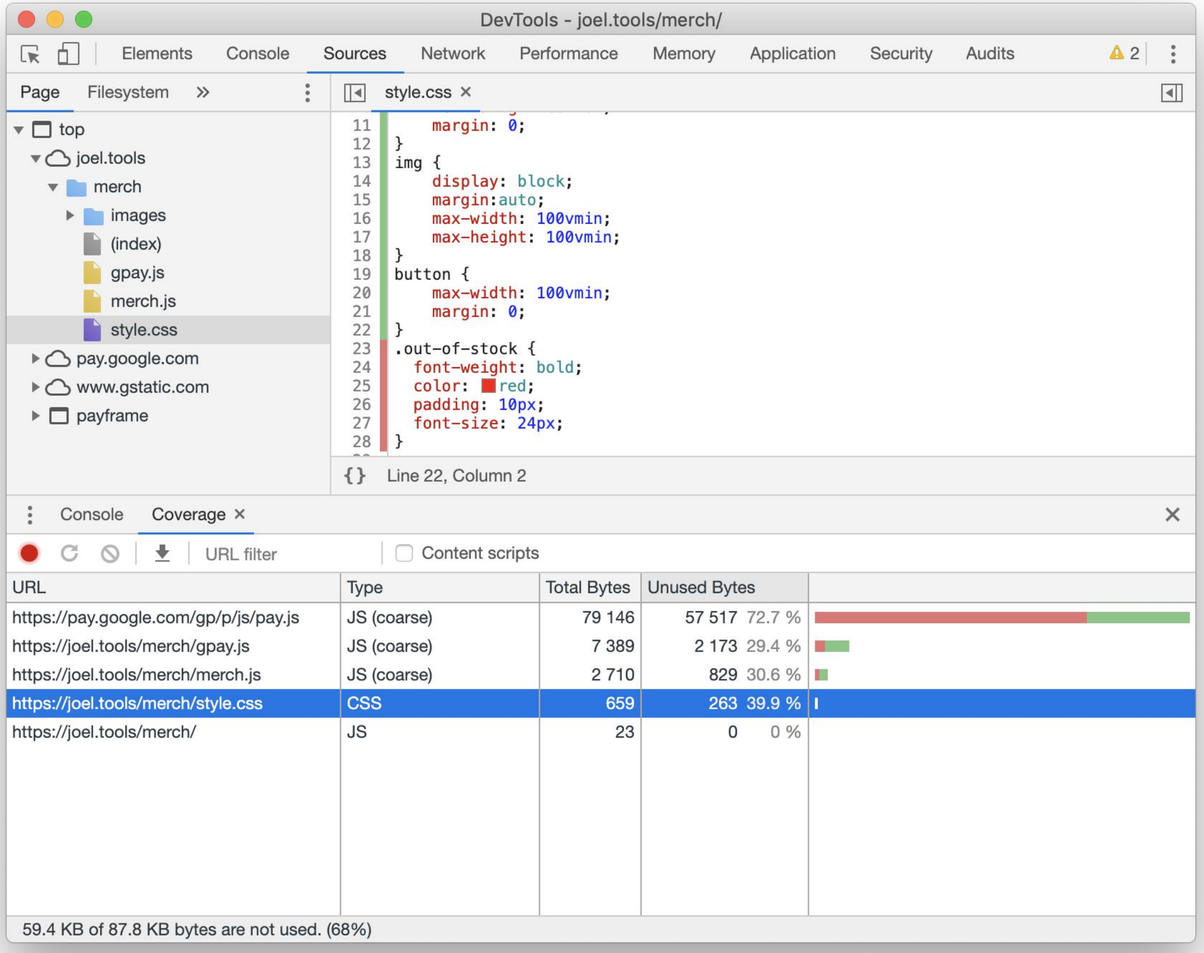
HEISENBUG

```
await page.goto('http://joel.tools/merch/');
await page.waitForSelector('select');
await page.select('select', 'mug');
```

```
await page.tracing.start({ path: './trace.json' });

await page.goto('http://joel.tools/merch/');
await page.waitForSelector('select');
await page.select('select', 'mug');
```

```
await page.tracing.start({ path: './trace.json' });

await page.goto('http://joel.tools/merch/');
await page.waitForSelector('select');
await page.select('select', 'mug');

await page.tracing.stop();
```

Code Coverage



HEISENBUG

Code Coverage

👉 `page.coverage.startCSSCoverage()`

👉 `page.coverage.stopCSSCoverage()`

👉 `page.coverage.startJSCoverage()`

👉 `page.coverage.stopJSCoverage()`

Accessibility API

👉 `page.accessibility.snapshot()`

👉 returns “**Chrome Accessibility Tree**”

👉 foundation for all assistive technologies

```
const snapshot = await page.accessibility.snapshot();
console.log(snapshot);

{ role: 'WebArea',
  name: 'Joel Merch',
  children: [
     { role: 'combobox', name: '', value: 'Shirt', children: [Array] },
     { role: 'heading', name: 'Shirt', level: 1 },
     { role: 'img', name: 'Shirt' },
     { role: 'text', name: 'This shirt is the height of fashion.' },
     { role: 'text', name: '$50' },
     { role: 'button', name: 'Buy with Google Pay', focused: true }
  ]
}
```

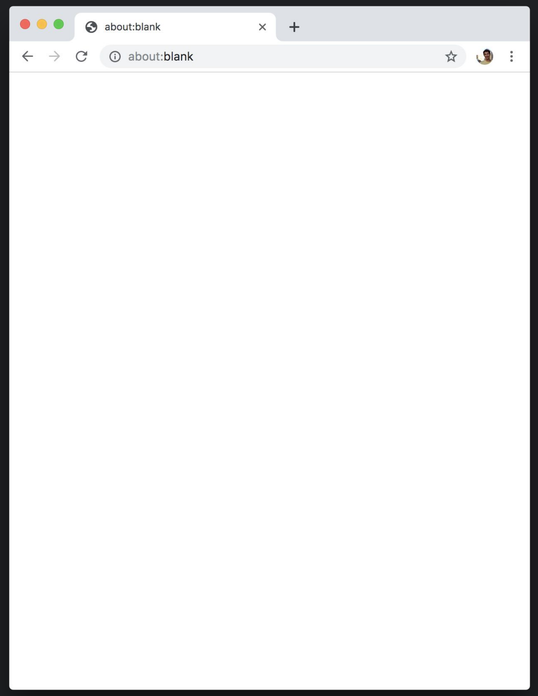
```
const snapshot = await page.accessibility.snapshot();
console.log(snapshot);

{ role: 'WebArea',
  name: 'Joel Merch',
  children: [
     { role: 'combobox', name: '', value: 'Shirt', children: [Array] },
     { role: 'heading', name: 'Shirt', level: 1 },
     { role: 'img', name: 'Shirt' },
     { role: 'text', name: 'This shirt is the height of fashion.' },
     { role: 'text', name: '$50' },
     { role: 'button', name: 'Buy with Google Pay', focused: true }
  ]
}
```

```
const snapshot = await page.accessibility.snapshot();
console.log(snapshot);

{ role: 'WebArea',
  name: 'Joel Merch',
  children: [
     { role: 'combobox', name: '', value: 'Shirt', children: [Array] },
     { role: 'heading', name: 'Shirt', level: 1 },
     { role: 'img', name: 'Shirt' },
     { role: 'text', name: 'This shirt is the height of fashion.' },
     { role: 'text', name: '$50' },
     { role: 'button', name: 'Buy with Google Pay', focused: true }
  ]
}
```

```
const snapshot = await page.accessibility.snapshot();
console.log(snapshot);

{ role: 'WebArea',
  name: 'Joel Merch',
  children: [
     { role: 'combobox', name: '', value: 'Shirt', children: [Array] },
     { role: 'heading', name: 'Shirt', level: 1 },
     { role: 'img', name: 'Shirt' },
     { role: 'text', name: 'This shirt is the height of fashion.' },
     { role: 'text', name: '$50' },
     { role: 'button', name: 'Buy with Google Pay', focused: true }
  ]
}
```

```
it('should have a well named pay button', async () => {
  // Test goes here




});
```

```
it('should have a well named pay button', async () => {
  const page = await context.newPage();




});
```

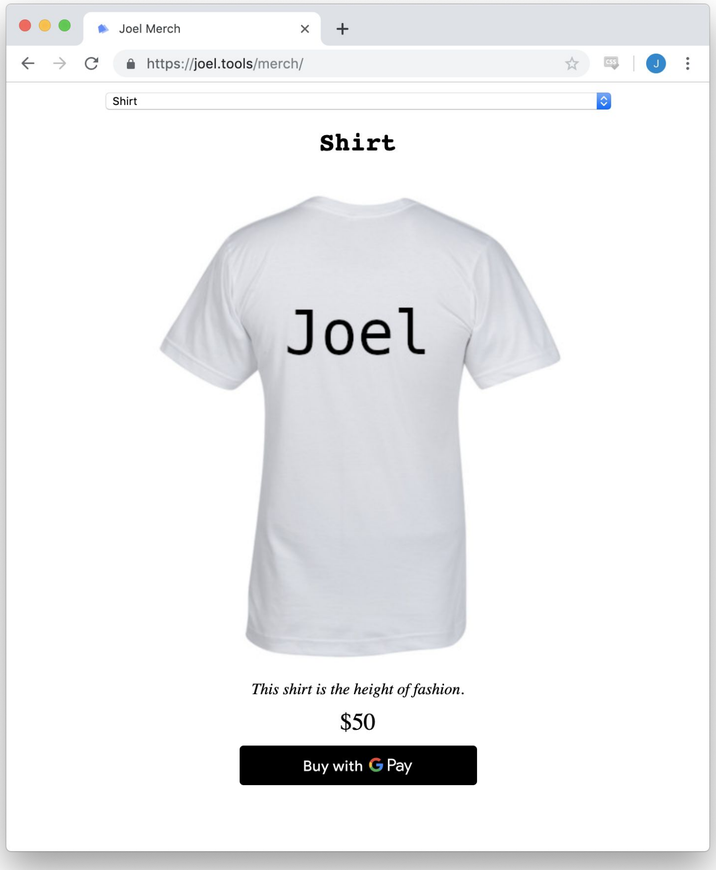


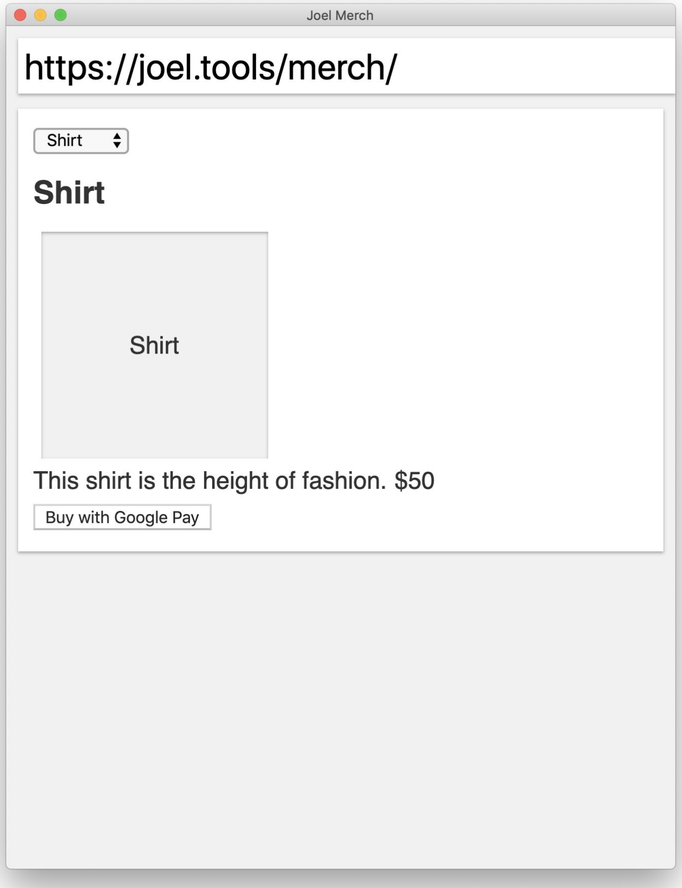
```
it('should have a well named pay button', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch');




});
```

```
it('should have a well named pay button', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch');
  const button = await page.$('button');



});
```

```
it('should have a well named pay button', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch');
  const button = await page.$('button');
  const buttonAX = await page.accessibility.snapshot({
    root: button,
  });

});
```

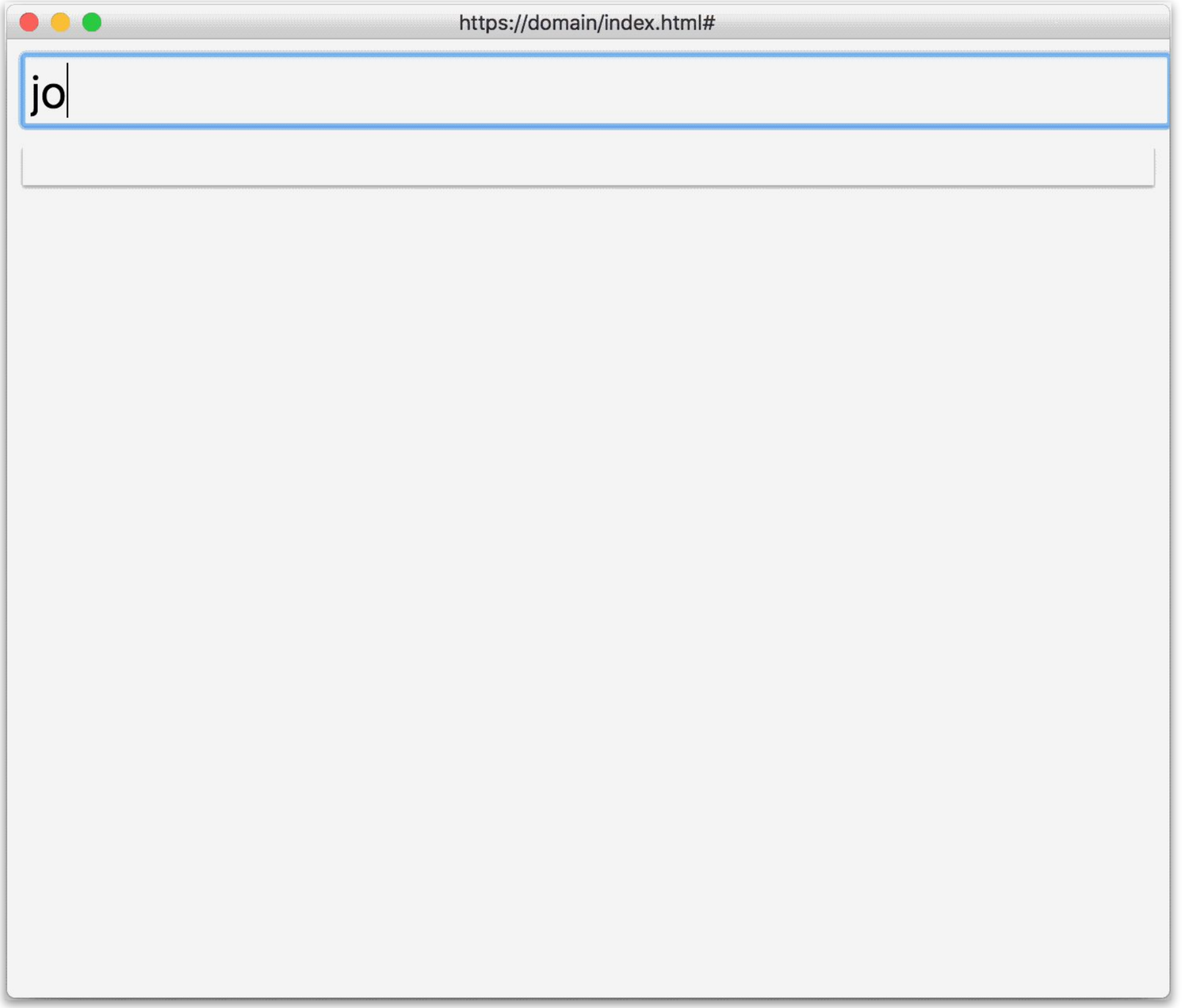
```
it('should have a well named pay button', async () => {
  const page = await context.newPage();
  await page.goto('https://joel.tools/merch');
  const button = await page.$('button');
  const buttonAX = await page.accessibility.snapshot({
    root: button,
  });
  expect(buttonAX.name).toBe('Buy with Google Pay');
});
```


Joel Merch
https://joel.tools/merch/
Shirt
Shirt
Joel
This shirt is the height of fashion.
$50
Buy with G Pay


Joel Merch
https://joel.tools/merch/
Shirt
Shirt
Shirt
This shirt is the height of fashion. $50
Buy with Google Pay


Puppeteer

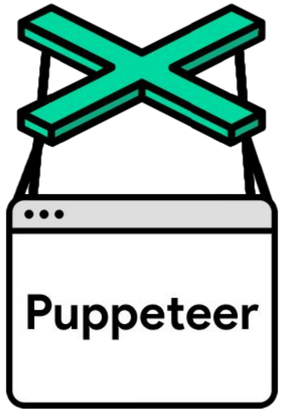
HEISENBUG



HEISENBUG

Accessibility Visualizer

👉 github.com/GoogleChromeLabs/html-a11y-renderer



github.com/GoogleChrome/puppeteer

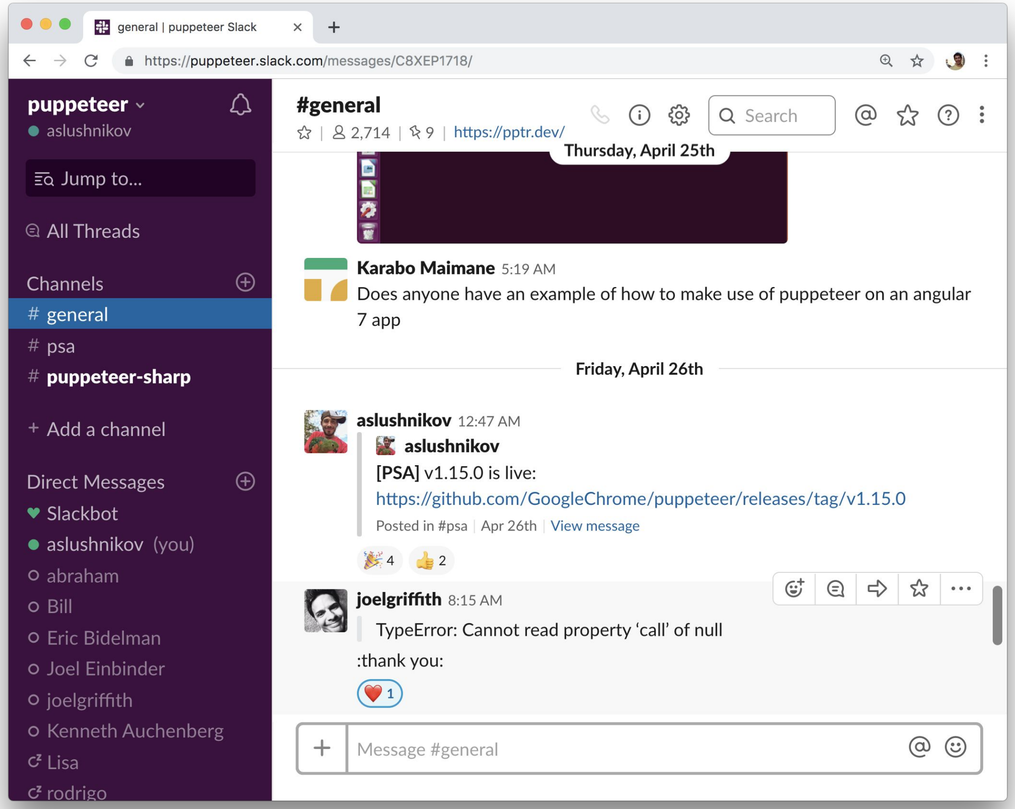
Google™

NETFLIX

HEISENBUG

Community 👫

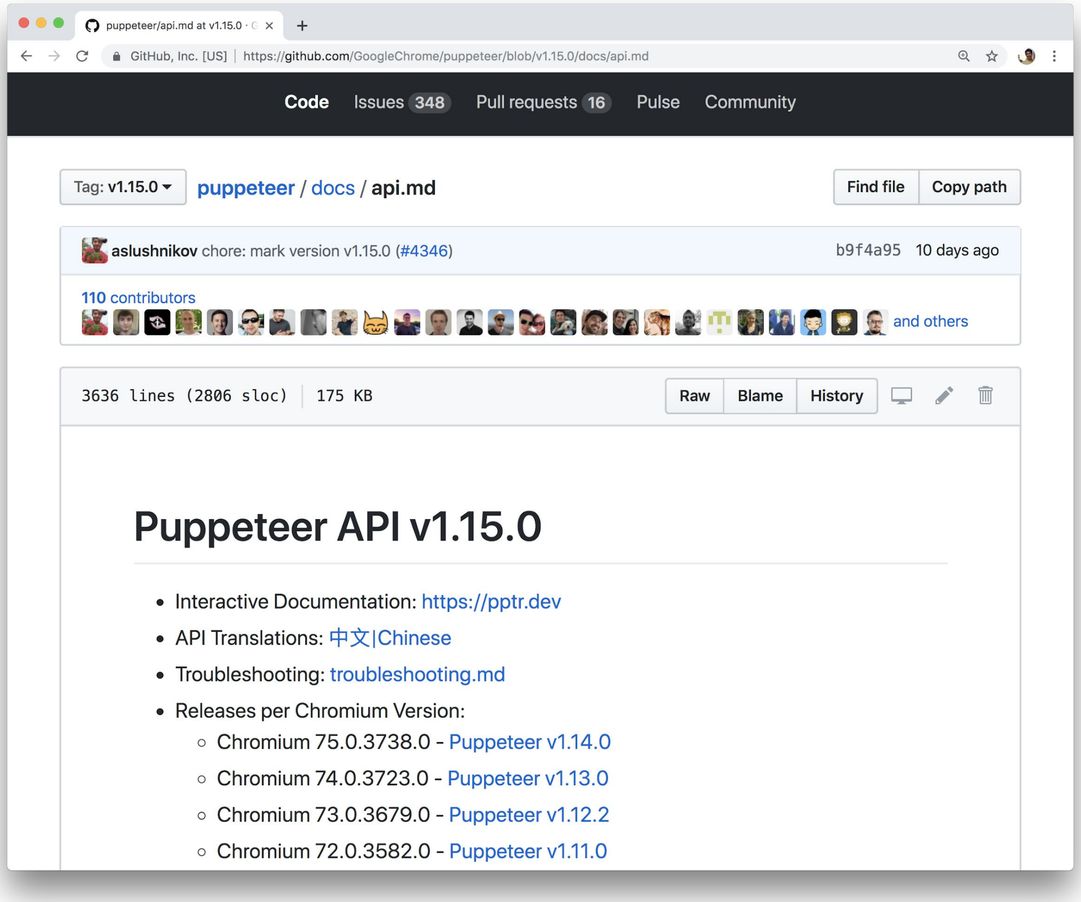
👉 **2700+** folks on puppeteer.slack.com



HEISENBUG

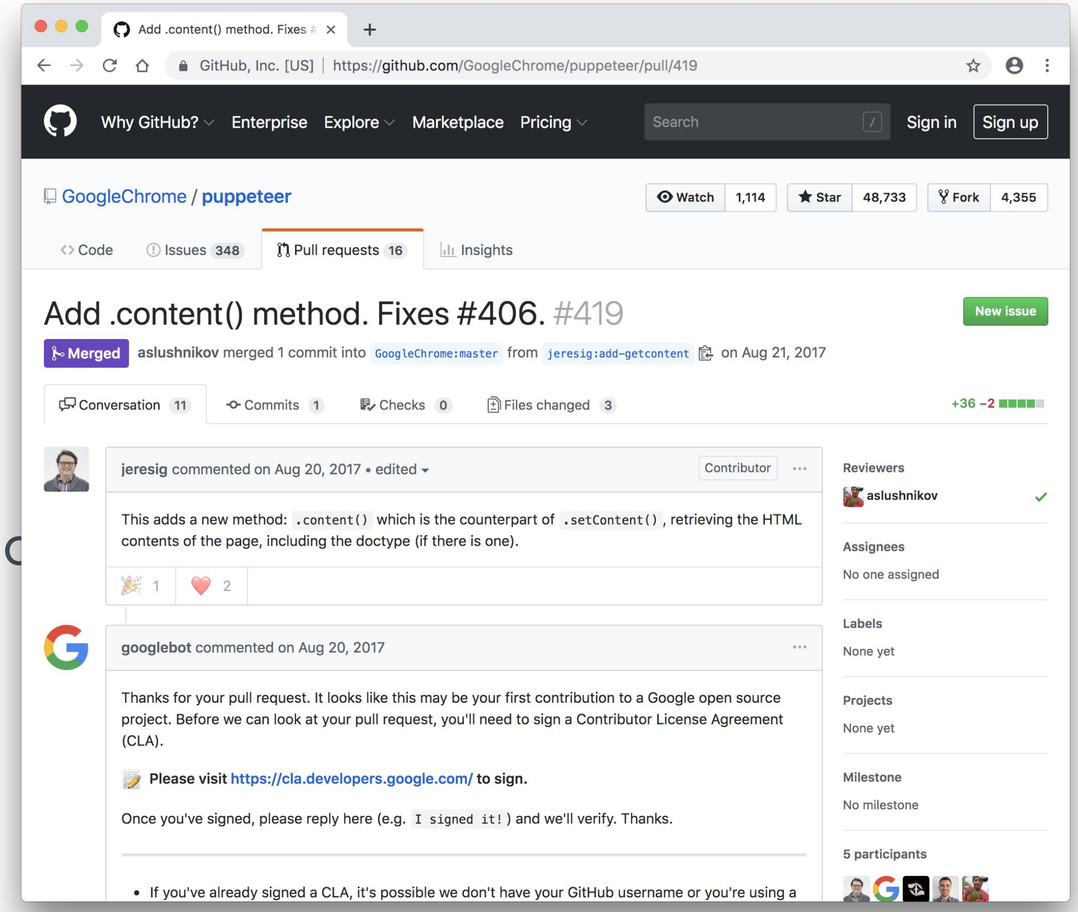
Community 👫

👉 **100+** people contributed to api docs

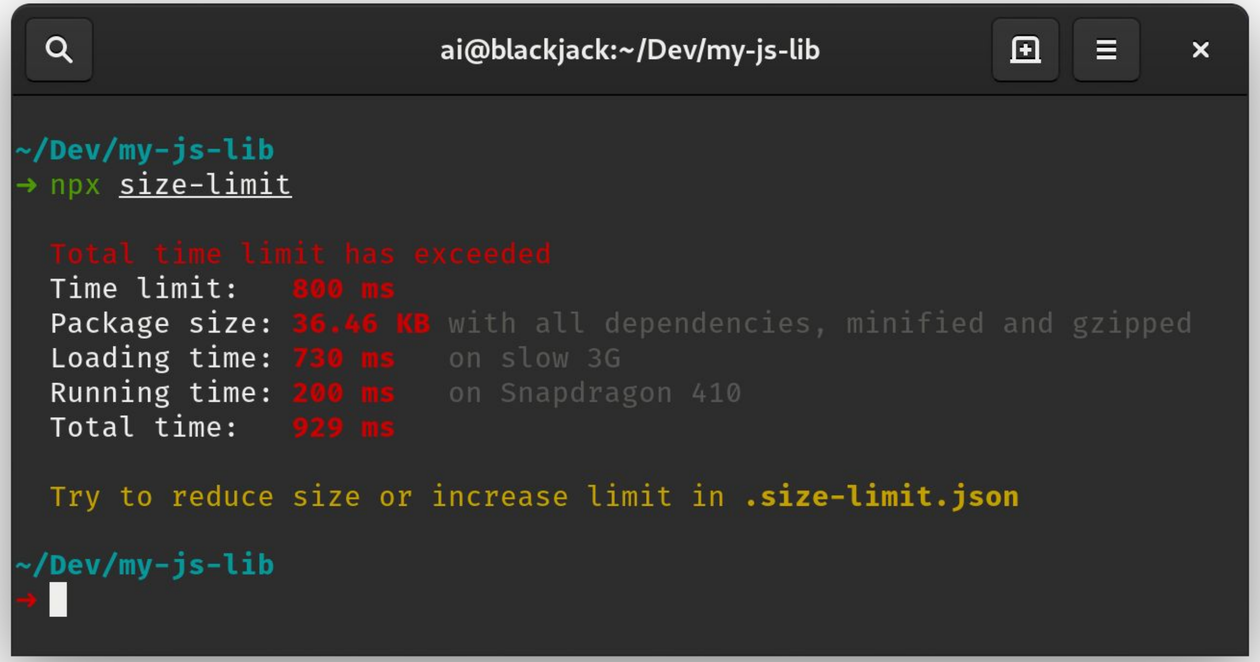


Community 👫

👉 **20%** of patches to `//lib` are community c

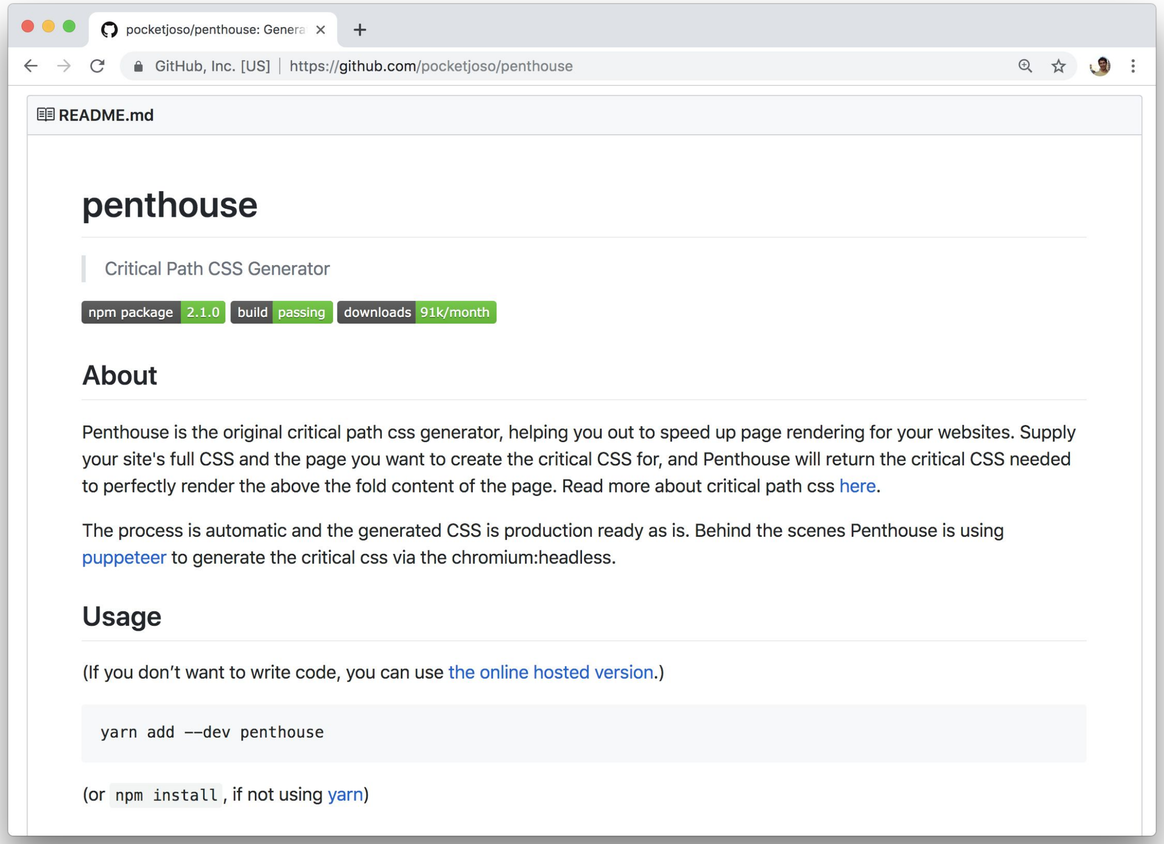


HEISENBUG



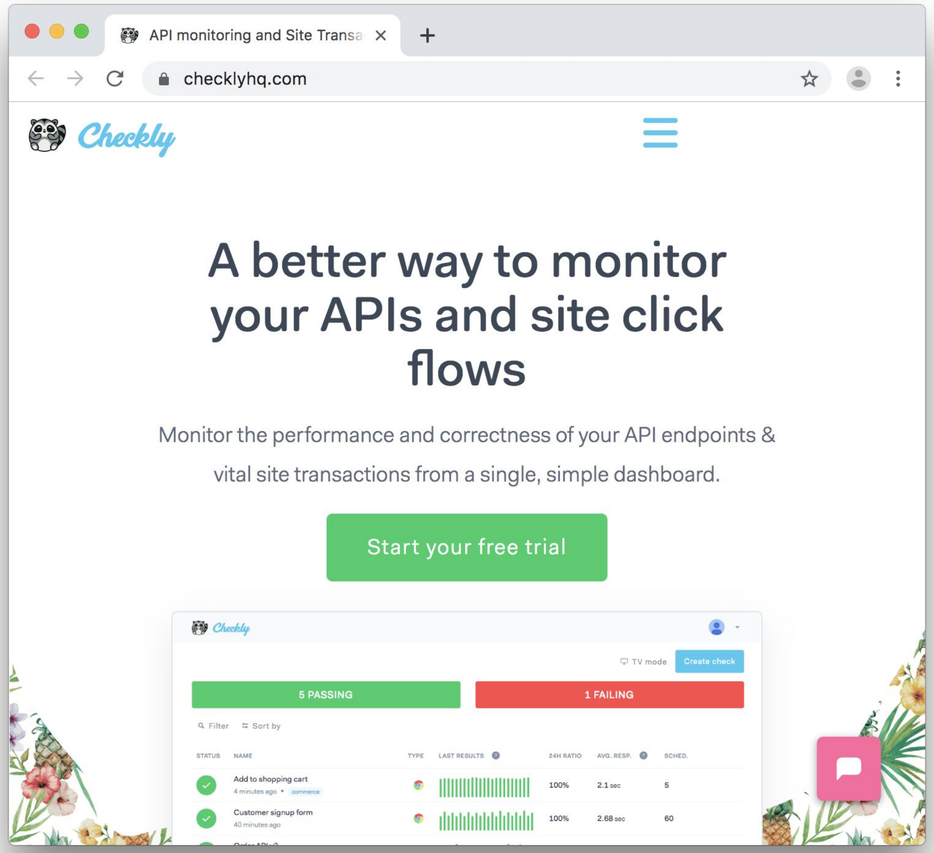
github.com/ai/size-limit

HEISENBUG



github.com/pocketjoso/penthouse

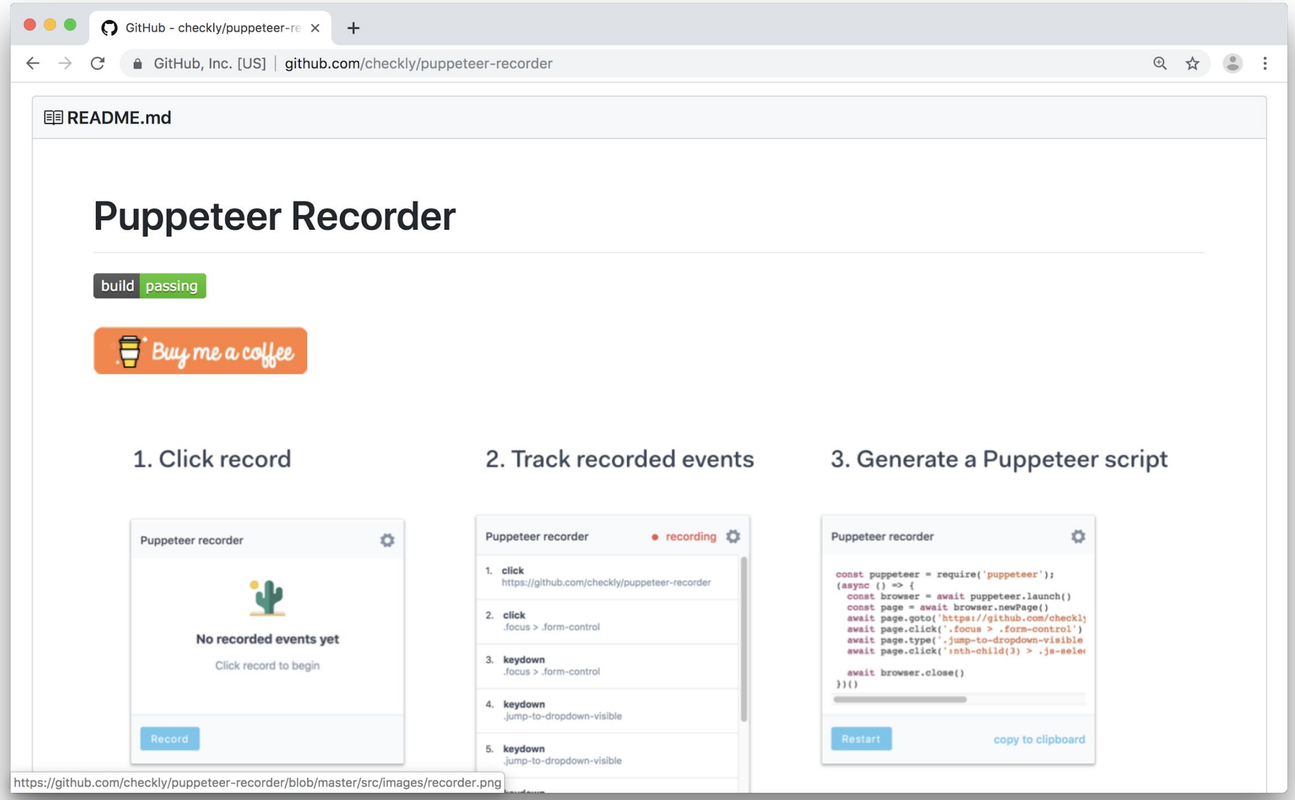
HEISENBUG



checklyhq.com

Puppeteer

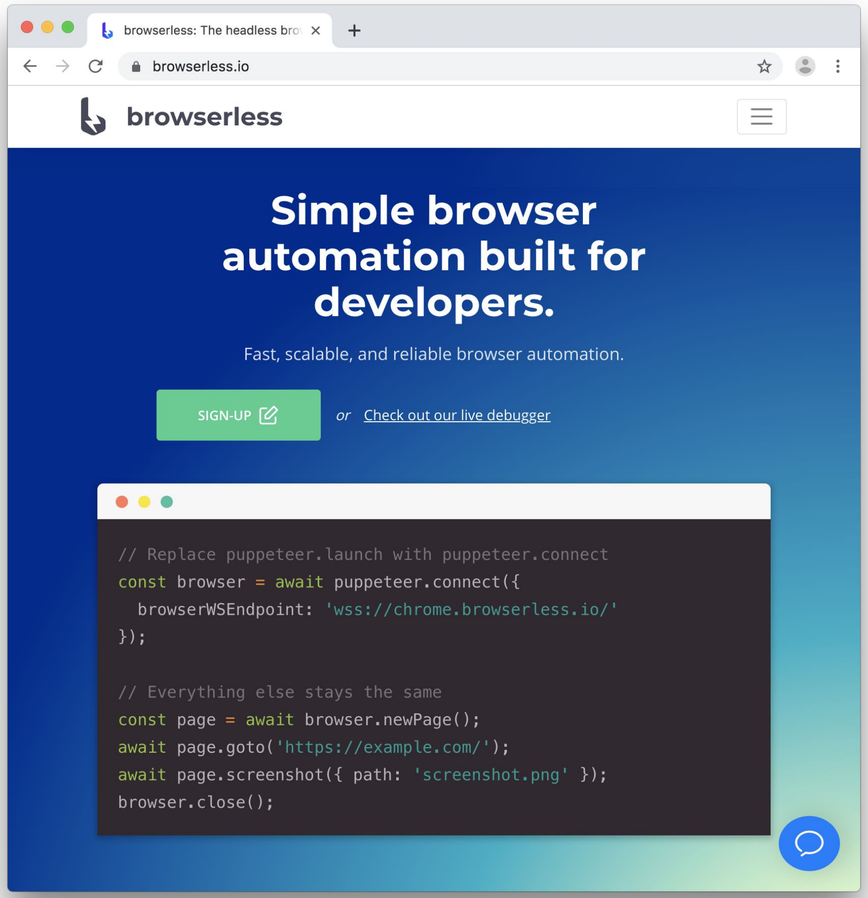
HEISENBUG



github.com/checkly/puppeteer-recorder

Puppeteer

HEISENBUG



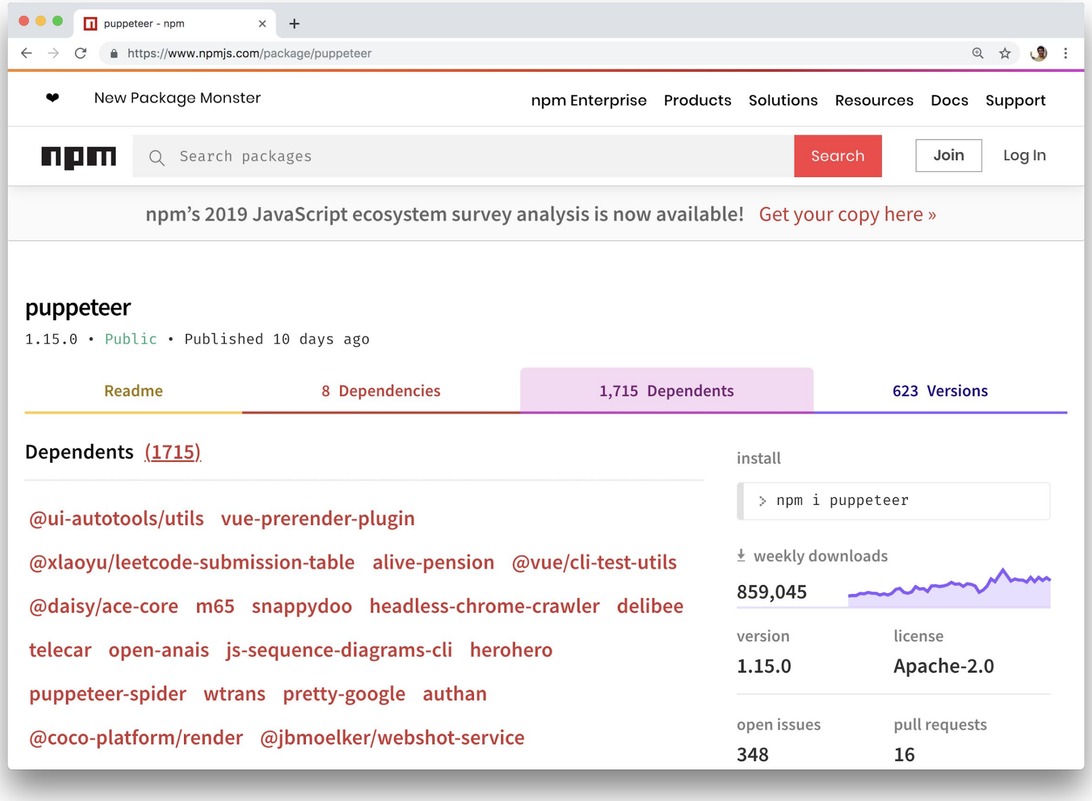
browserless.io

Puppeteer

HEISENBUG

Community 👫

👉 **1700+** packages on NPM



HEISENBUG

TL;DR

👉 The **NodeJS** API for Chrome
👉 Experimental **Firefox** support
👉 **Test** the Web with Puppeteer

Thank you!

Andrey Lushnikov
Google, @aslushnikov