How to set max viewport in Puppeteer?

Asked 3 years, 4 months ago   Active 1 year, 2 months ago   Viewed 70k times

▲

55

▼

🔖

15

🕐

When I run a new page, I must specify size of the viewport using the `setViewport` function:

```
await page.setViewport({
    width: 1920,
    height: 1080
})
```

I want use max viewport.

How can I make the viewport resizable according to the window size?

`javascript` `node.js` `google-chrome-devtools` `viewport` `puppeteer`

Share Improve this question Follow

edited Sep 28 '18 at 19:04
Grant Miller
22.9k ● 15 ● 128 ● 146

asked Sep 28 '18 at 10:23
fehif
551 ● 1 ● 4 ● 4

8 Answers

Active | Oldest | Score

▲

91

▼

🕐

I may be very late on this. Nevertheless for others, try:

```
const browser = await puppeteer.launch({defaultViewport: null});
```

Set the `defaultViewport` option to `null` as above to disable the 800x600 resolution. It takes the max resolution then.

Share Improve this answer Follow

edited Apr 4 '19 at 19:29
gdoron is supporting Monica
141k ● 53 ● 280 ● 352

answered Mar 22 '19 at 14:59
n00b_g33k
921 ● 5 ● 6

1   @gdoron It is mentioned in their documentation, puppeteer.launch(options) . See `defaultViewport` option, it clearly states that setting this to `null` , it disables the default viewport + it worked for me. – n00b_g33k Apr 3 '19 at 8:24 ✏

1   Worked for me with: const browser = await puppeteer.launch({ headless: false, defaultViewport: null }); – jenkizenki Apr 17 '19 at 14:38 ✏

When disabling headless mode and visually launching Chromium, this confirms that the viewport uses its maximum allowed size. Together with pdf option `preferCSSPageSize: true` this allows for much easier styling of a pdf. – G_V Jun 18 '19 at 13:03

In my case both your answer and the one of @Grant Miller (namely using the args to set the window size) have been the solution – EuberDeveloper yesterday

---

▲

48

▼

🕐

You can pass the `--window-size` flag as an argument to `puppeteer.launch()` to change the window size to your desired `width` and `height`.

Then you can call the Chrome Devtools Protocol method `Emulation.clearDeviceMetricsOverride` to clear the overridden device metrics (including the default 800 x 600 viewport).

This will cause the viewport to match the window size (when taking screenshots, etc).

```
const browser = await puppeteer.launch({
  args: [
    '--window-size=1920,1080',
  ],
});

const page = await browser.newPage();

await page._client.send('Emulation.clearDeviceMetricsOverride');

await page.screenshot({
  path: 'example.png', // Image Dimensions : 1920 x 1080
});
```

> **Note:** `page.viewport()` will still return `{ width: 800, height: 600 }`, and the only way to reliably change the values of these properties is to use `page.setViewport()` .

See the complete example below:

```
'use strict';

const puppeteer = require('puppeteer');

(async () => {
  /* ==========================================================
      Prerequisite: Set Window size
  ========================================================== */

  const browser = await puppeteer.launch({
    args : [
      '--window-size=1920,1080',
    ],
  });

  const page = await browser.newPage();

  await page.goto('https://www.example.com/');

  /* ==========================================================
      Case 1: Default Viewport
  ========================================================== */

  console.log('Case 1 - Width  :', page.viewport().width);  // Width  : 800
  console.log('Case 1 - Height :', page.viewport().height); // Height : 600

  await page.screenshot({
    path: 'image-1.png', // Image Dimensions : 800 x 600
  });

  /* ==========================================================
      Case 2: Clear Overridden Device Metrics
  ========================================================== */

  await page._client.send('Emulation.clearDeviceMetricsOverride');

  console.log('Case 2 - Width  :', page.viewport().width);  // Width  : 800
  console.log('Case 2 - Height :', page.viewport().height); // Height : 600

  await page.screenshot({
    path: 'image-2.png', // Image Dimensions : 1920 x 1080
  });

  /* ==========================================================
      Case 3: Manually Set Viewport
  ========================================================== */

  await page.setViewport({
    width: 1920,
    height: 1080,
  });

  console.log('Case 3 - Width  :', page.viewport().width);  // Width  : 1920
  console.log('Case 3 - Height :', page.viewport().height); // Height : 1080

  await page.screenshot({
    path: 'image-3.png', // Image Dimensions : 1920 x 1080
  });

  /* ==========================================================
      Case 4: Revert Back to Default Viewport
  ========================================================== */

  await page.setViewport({
    width: 800,
    height: 600,
  });

  console.log('Case 4 - Width  :', page.viewport().width);  // Width  : 800
  console.log('Case 4 - Height :', page.viewport().height); // Height : 600

  await page.screenshot({
    path: 'image-4.png', // Image Dimensions : 800 x 600
  });

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

edited Aug 30 '19 at 3:27

answered Sep 28 '18 at 19:03
Grant Miller
22.9k ● 15 ● 128 ● 146

...rDeviceMetricsOverride'); makes `--window-size` work – datdinhquoc Feb 7 '21 at

...ent screen resolution. Seems to only work with `headless: false`

answered Nov 21 '18 at 17:34

```
const browser = await puppeteer.launch( {"headless": false, args: ['--start-
maximized'] } );
const page = await browser.newPage();
await page.setViewport({width:0, height:0});
```

14

the " const browser = ..." line maximizes your chrome window. But note that the page is where the viewport needs to be set and it would still be at the default size when you create the page.

When you set the viewport with width and height as "0", the page viewport size becomes equal to size of the browser.

Share Improve this answer Follow

edited May 9 '19 at 18:33

answered May 8 '19 at 17:43

Rahul Singh
345 ● 3 ● 11

12

For me the combination of `defaultViewport: null` and `args: ['--start-maximized']` gave me fullscreen with view fitting to screensize so:

```
browser = await puppeteer.launch({
            headless: false,
            args: [
                '--start-maximized',
            ],
            defaultViewport: null,
        });
```

Share Improve this answer Follow

answered Mar 4 '20 at 18:08

Zubeir
31 ● 3 ● 5

10

I had the same problem and what worked for me is this.

```
const browser = await puppeteer.launch({
    headless: false,
    args: ['--window-size=1920,1080'],
    defaultViewport: null
    });
```

Share Improve this answer Follow

answered Aug 21 '20 at 10:59

AMMA
145 ● 1 ● 8

1   use `await page._client.send('Emulation.clearDeviceMetricsOverride');` for when headless=true – datdinhquoc Feb 7 '21 at 12:28

8

It's probably worth mentioning that if you combine `puppeteer.launch({defaultViewport: null})` with `puppeteer.connect()`, again you need to pass `{defaultViewport: null}}`, otherwise the viewport is adjusted back to default size. So in this case, use:

```
await puppeteer.connect({browserWSEndpoint: ws, defaultViewport: null})
```

Share Improve this answer Follow

answered Jul 31 '19 at 15:09

Vangelisz Ketipisz
695 ● 1 ● 6 ● 8

1

Here is a way to do it at runtime by calling `Page.setViewport()` in headful and `Browser.setWindowBounds()` in headless via a Chrome Devtools Protocol session:

```
async function setWindowSize(page, width, height) {
  if(headless) {
    const session = await page.target().createCDPSession();
    const {windowId} = await session.send('Browser.getWindowForTarget');
    await session.send('Browser.setWindowBounds', {windowId, bounds: {width:
width, height: height}});
    await session.detach();
  } else {
    await page.setViewport({width: width, height: height});
  }
}
```

See my comment on GitHub for more info.

Share Improve this answer Follow

answered Nov 17 '20 at 3:12

Zack Morris
4,404 ● 2 ● 51 ● 79