

What's New In DevTools (Chrome 59)



By Kayce Basques

Technical Writer for Chrome DevTools

Welcome to another installment of the DevTools release notes. Watch the video below or read on to see what's new in Chrome DevTools in Chrome 59!

Note: You can check which version of Chrome you're running at <chrome://help>.

Highlights

- **CSS and JS code coverage**. Find unused CSS and JS with the new Coverage tab.
- **Full-page screenshots**. Take a screenshot of the entire page, from the top of the viewport to the bottom.
- **Block requests**. Manually disable individual requests in the Network panel.
- **Step over async await**. Step through async functions predictably.

- **Unified Command Menu.** Execute commands and open files from the newly-unified Command Menu.

New features

CSS and JS code coverage

Find unused CSS and JS code with the new **Coverage** tab. When you load or run a page, the tab tells you how much code was used, versus how much was loaded. You can reduce the size of your pages by only shipping the code that you need.

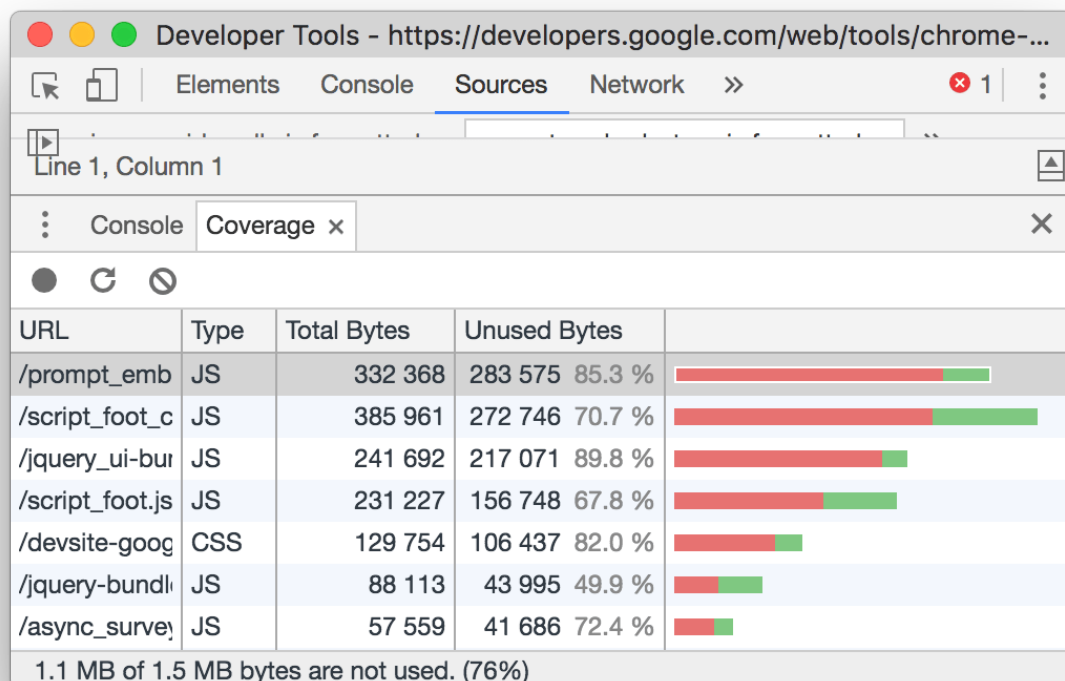


Figure 1. The Coverage tab

Clicking on a URL reveals that file in the **Sources** panel with a breakdown of which lines of code executed.

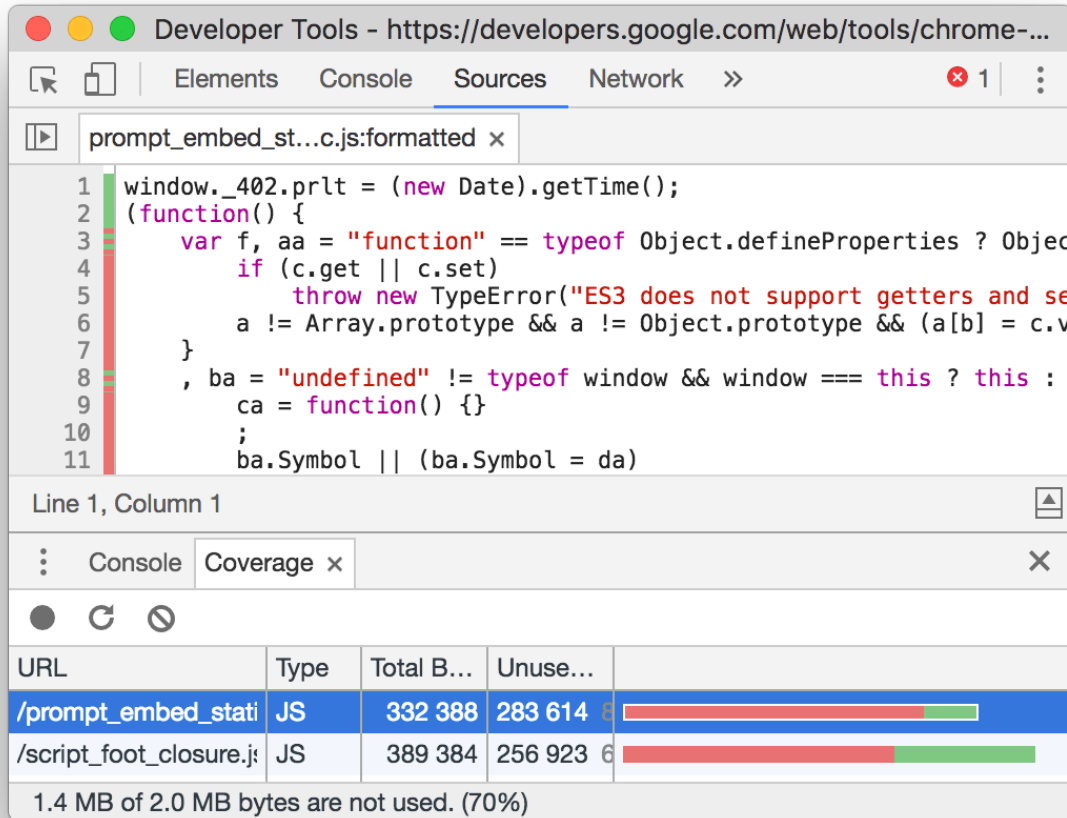


Figure 2. A breakdown of code coverage in the **Sources** panel

Each line of code is color-coded:

- Solid green means that line of code executed.
- Solid red means it did not execute.
- A line of code that is both red and green, such as line 3 in **Figure 2**, means that only some code on that line executed. For example, a ternary expression like `var b = (a > 0) ? a : 0` is colored both red and green.

Note: The color-coding is likely to change in future Chrome releases.

To open the **Coverage** tab:

1. Open the Command Menu.
2. Start typing **Coverage** and select **Show Coverage**.

Full-page screenshots

Check out the video below to learn how to take a screenshot from the top of the page, all the way to the bottom.

Block requests

Want to see how your page behaves when a particular script, stylesheet, or other resource isn't available? Right-click on the request in the **Network** panel and select **Block Request URL**. A new **Request blocking** tab pops up in the Drawer, which lets you manage blocked requests.

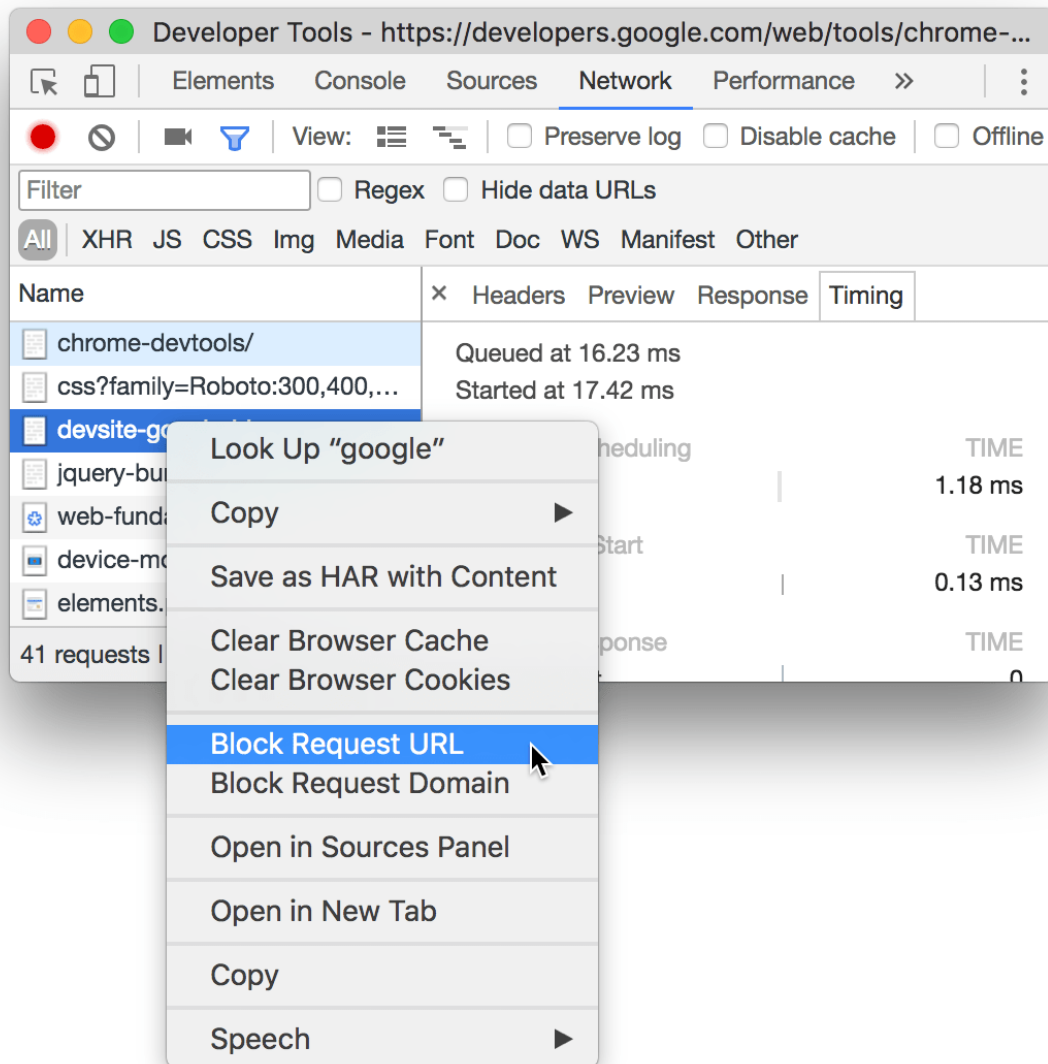


Figure 3. Block Request URL

Step over async await

Up until now, trying to step through code like the snippet below was a headache. You'd be in the middle of `test()`, stepping over a line, and then you'd get interrupted by the `setInterval()` code. Now, when you step through async code like `test()`, DevTools steps from the first to last line with consistency.

```
function wait(ms) {  
  return new Promise(r => setTimeout(r, ms)).then(() => "Yay");  
}  
  
// do some work in background.  
setInterval(() => 42, 200);
```



```
async function test() {
  debugger;
  const hello = "world";
  const response = await fetch('index.html');
  const tmp = await wait(1000);
  console.log(tmp);
  return hello;
}

async function runTest() {
  let result = await test();
  console.log(result);
}
```

P.S. want to level up your debugging skills? Check out these new-ish docs:

- [Get Started With Debugging JS](#)
- [Pause Your Code With Breakpoints](#)
- [JS Debugging Reference](#)

Changes

Unified Command Menu

When you open the [Command Menu](#) now, notice that your command is prepended with a greater-than character (>). This is because the Command Menu has been unified with the **Open File** menu, which is Command+O (Mac), or Control+O (Windows, Linux).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 2, 2018.