What's New In DevTools (Chrome 67)

By Kayce Basques
Technical Writer for Chrome DevTools

New features and major changes coming to DevTools in Chrome 67 include:

- Search across all network headers
- CSS variable value previews in the **Styles** pane
- Copy as fetch
- New audits, desktop configuration options, and viewing traces
- Stop infinite loops
- User Timing in the **Performance** tabs
- JavaScript VM instances clearly listed in the **Memory** panel
- The **Network** tab in the **Sources** panel has been renamed to the **Page** tab
- Dark theme updates
- Certificate transparency information in the **Security** panel
- Site isolation features in the **Performance** panel

Video version of the release notes:

Note: Check what version of Chrome you're running at `chrome://version`. If you're running an earlier version, these features won't exist. If you're running a later version, these features may have changed. Chrome auto-updates to a new major version about every 6 weeks.

Search across all network headers and responses

Open the **Network** panel then press Command+F (Mac) or Control+F (Windows, Linux, Chrome OS) to open the new **Network Search** pane. DevTools searches the headers and bodies of all network requests for the query that you provide.

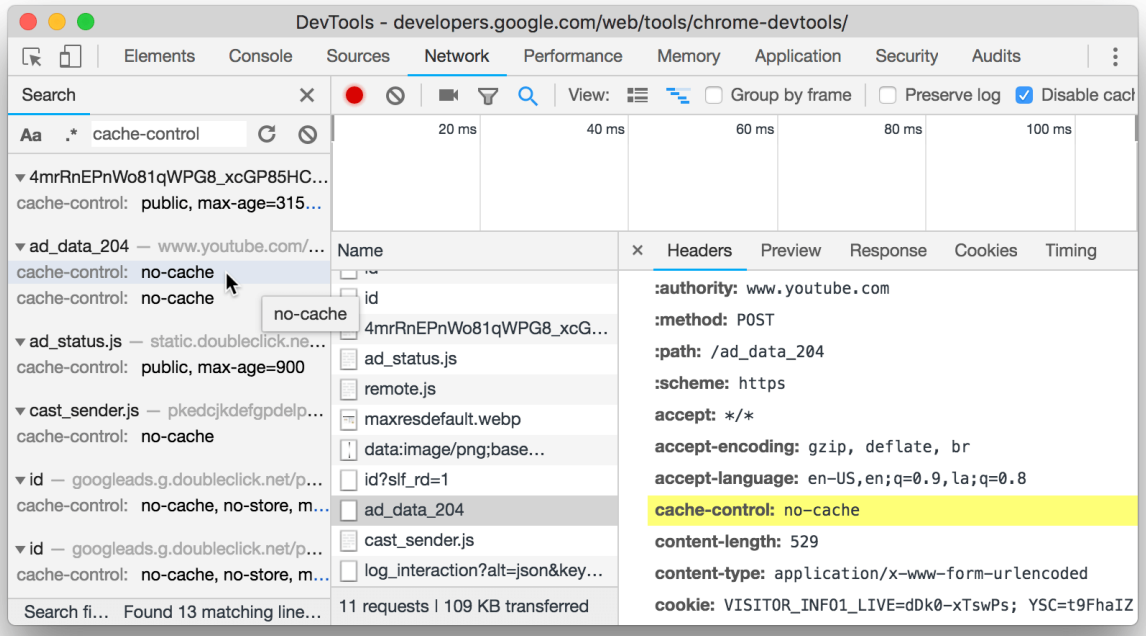

Figure 1. Searching for the text `cache-control` with the new Network Search pane

Click **Match Case** Aa to make your query case-sensitive. Click **Use Regular Expression** .* to show any results that match the pattern you provide. You don't need to wrap your RegEx in forward slashes.

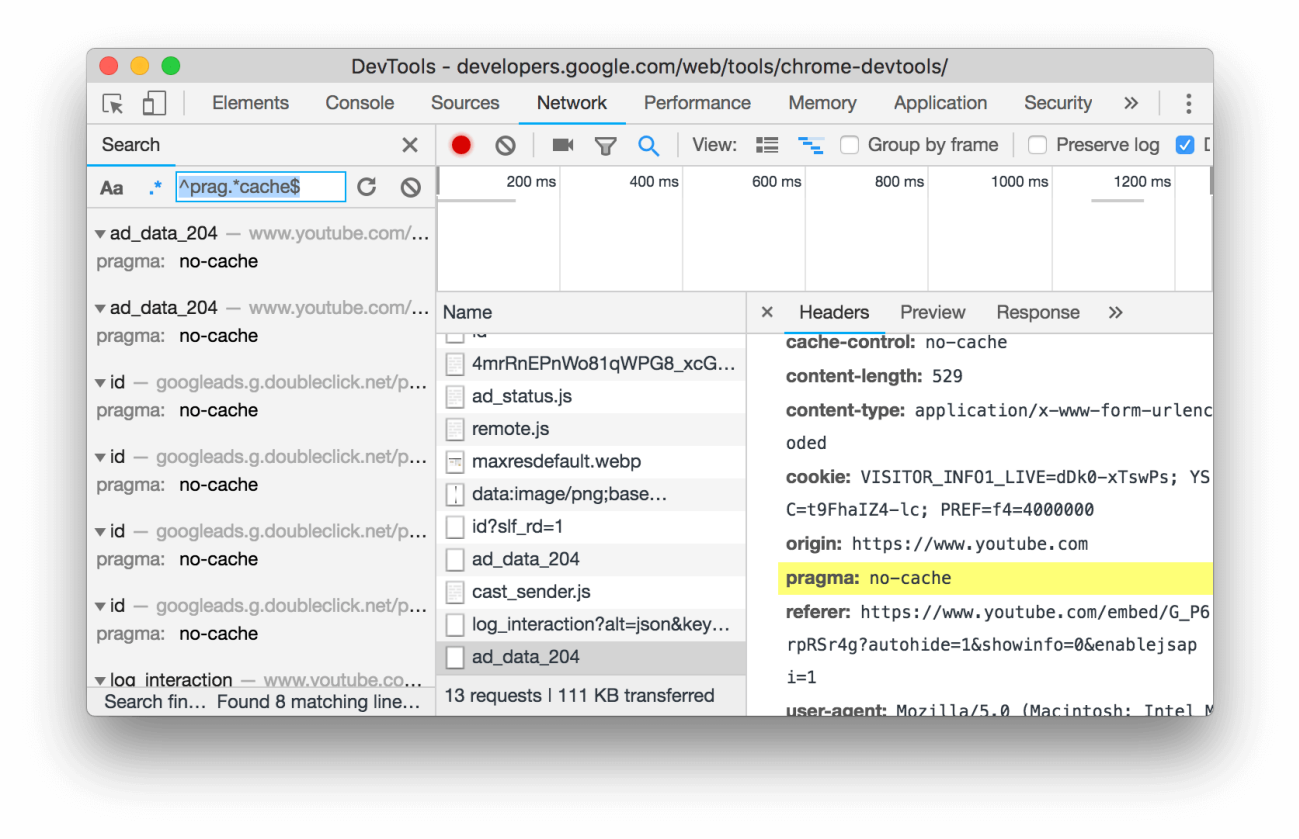

Figure 2. A regular expression query in the Network Search pane.

Search pane UI updates

The UI of the **Global Search** pane now matches the UI of the new **Network Search** pane. It now also pretty-prints results to aid scannability.

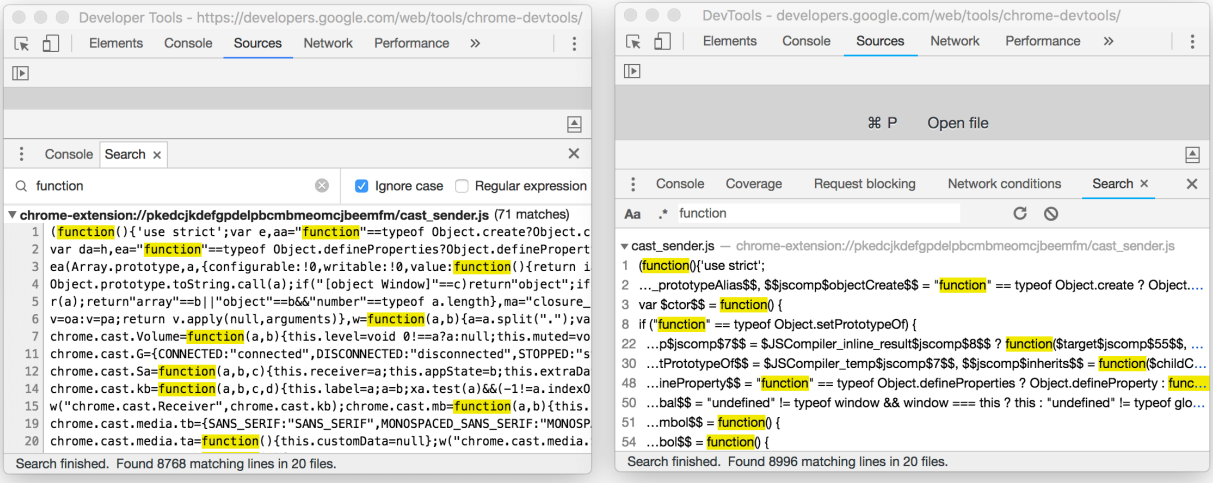

Figure 3. The old UI on the left, and the new UI on the right

Press Command+Option+F (Mac) or Control+Shift+F (Windows, Linux, Chrome OS) to open **Global Search**. You can also open it via the Command Menu.

CSS variable value previews in the **Styles** pane

When the value of a CSS color property, such as `background-color` or `color`, is set to a CSS variable, DevTools now shows a preview of that color.

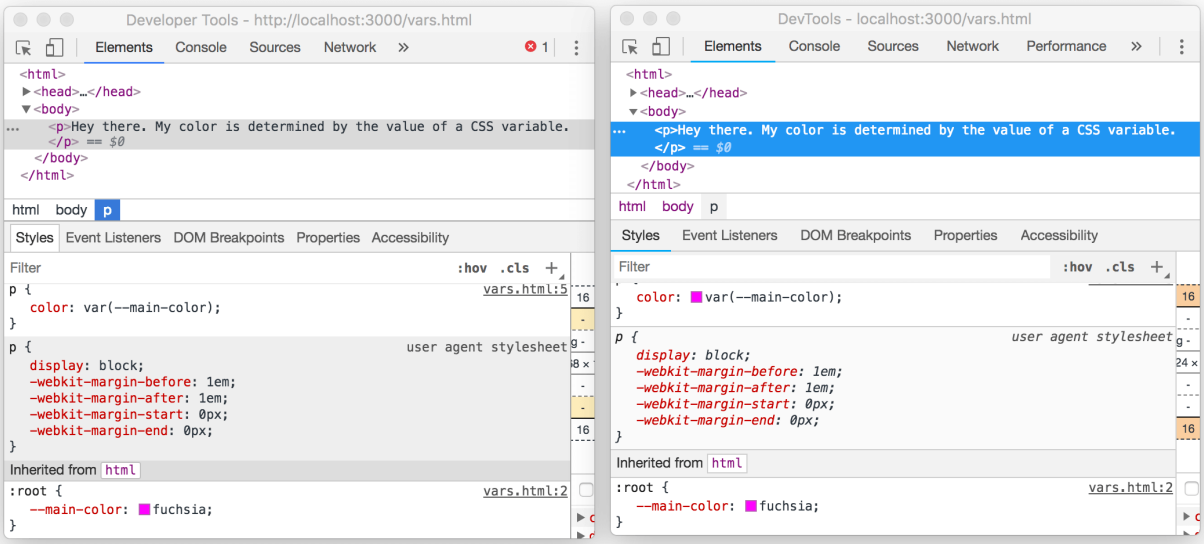

Figure 4. In the old UI on the left, there is no color preview next to `color: var(--main-color)`, whereas in the new UI on the right, there is

Copy as fetch

Right-click a network request then select **Copy** > **Copy As Fetch** to copy the `fetch()`-equivalent code for that request to your clipboard.

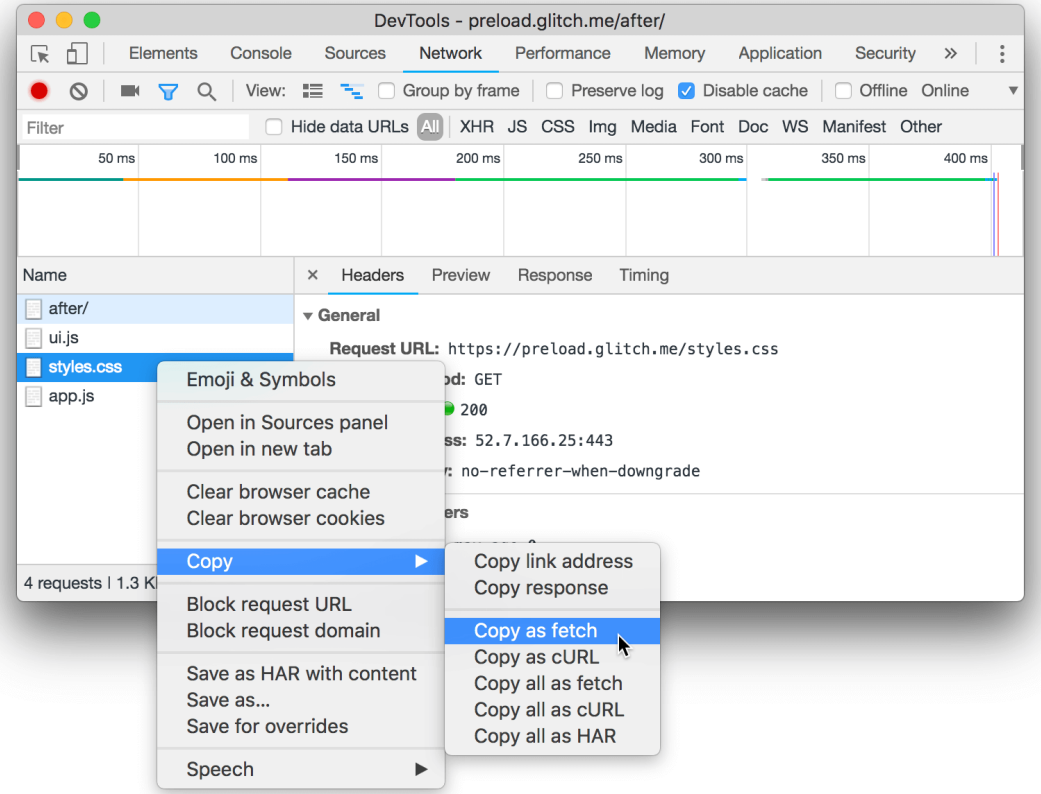

Figure 5. Copying the `fetch()`-equivalent code for a request

DevTools produces code like the following:

```
fetch("https://preload.glitch.me/styles.css", {
  "credentials": "omit",
  "headers": {},
  "referrer": "https://preload.glitch.me/after/",
  "referrerPolicy": "no-referrer-when-downgrade",
  "body": null,
  "method": "GET",
  "mode": "cors"
});
```

Audits panel updates

New audits

The **Audits** panel has 2 new audits, including:

- Preload key requests. Preloading requests can speed up page load time by giving hints to the browser to download resources that are important for your Critical Rendering Path as soon as possible.
- Avoid invisible text while webfonts are loading. Ensuring that text is visible while webfonts load makes the page more useful to users faster.

New configuration options

You can now configure the **Audits** panel to:

- Preserve desktop viewport and user agent settings. In other words, you can prevent the **Audits** panel from simulating a mobile device.
- Disable network and CPU throttling.
- Preserve storage, such as LocalStorage and IndexedDB, across audits.

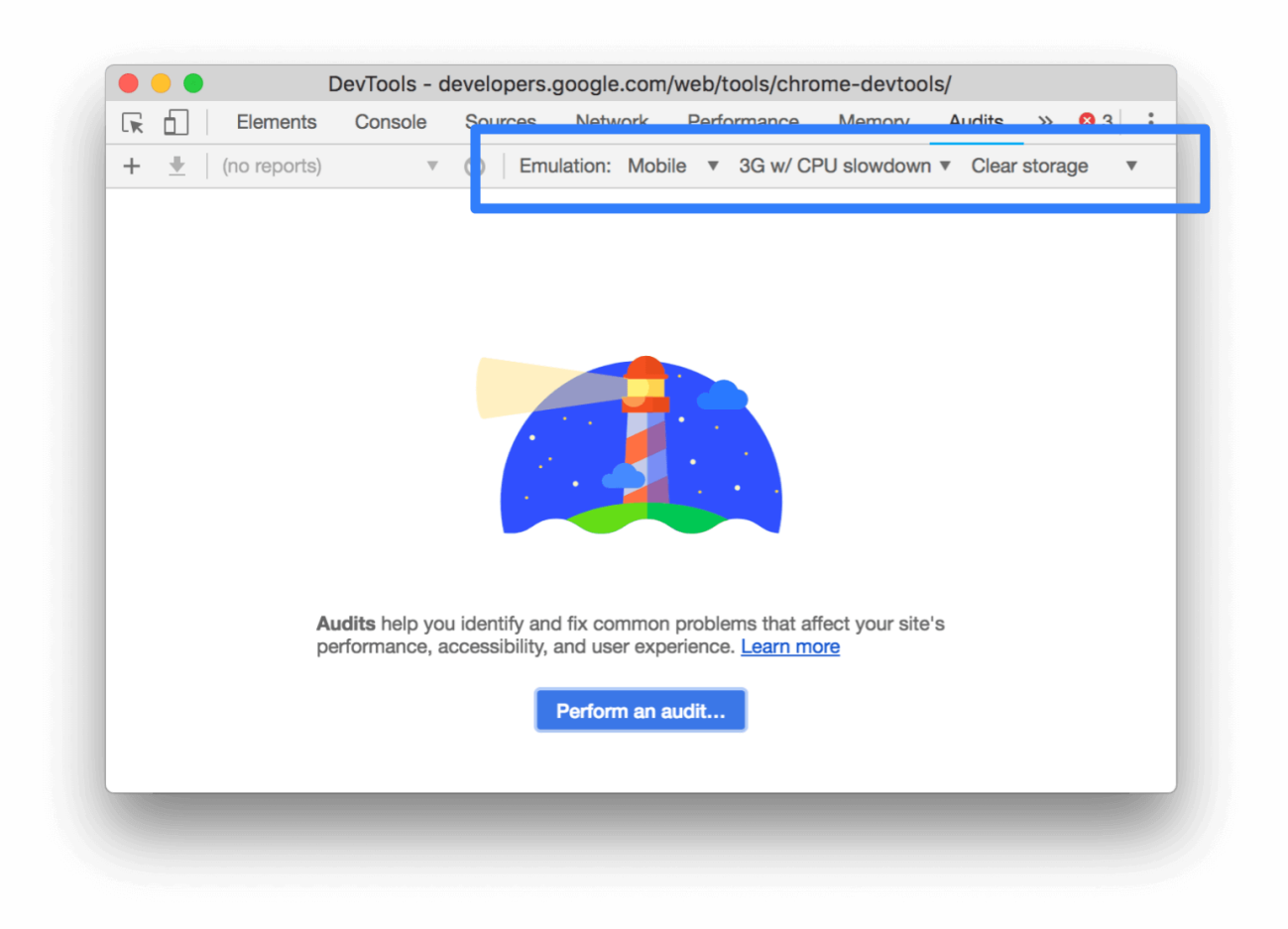

Figure 6. New audit configuration options

View traces

After auditing a page, click **View Trace** to view the load performance data that your audit is based off of in the **Performance** panel.

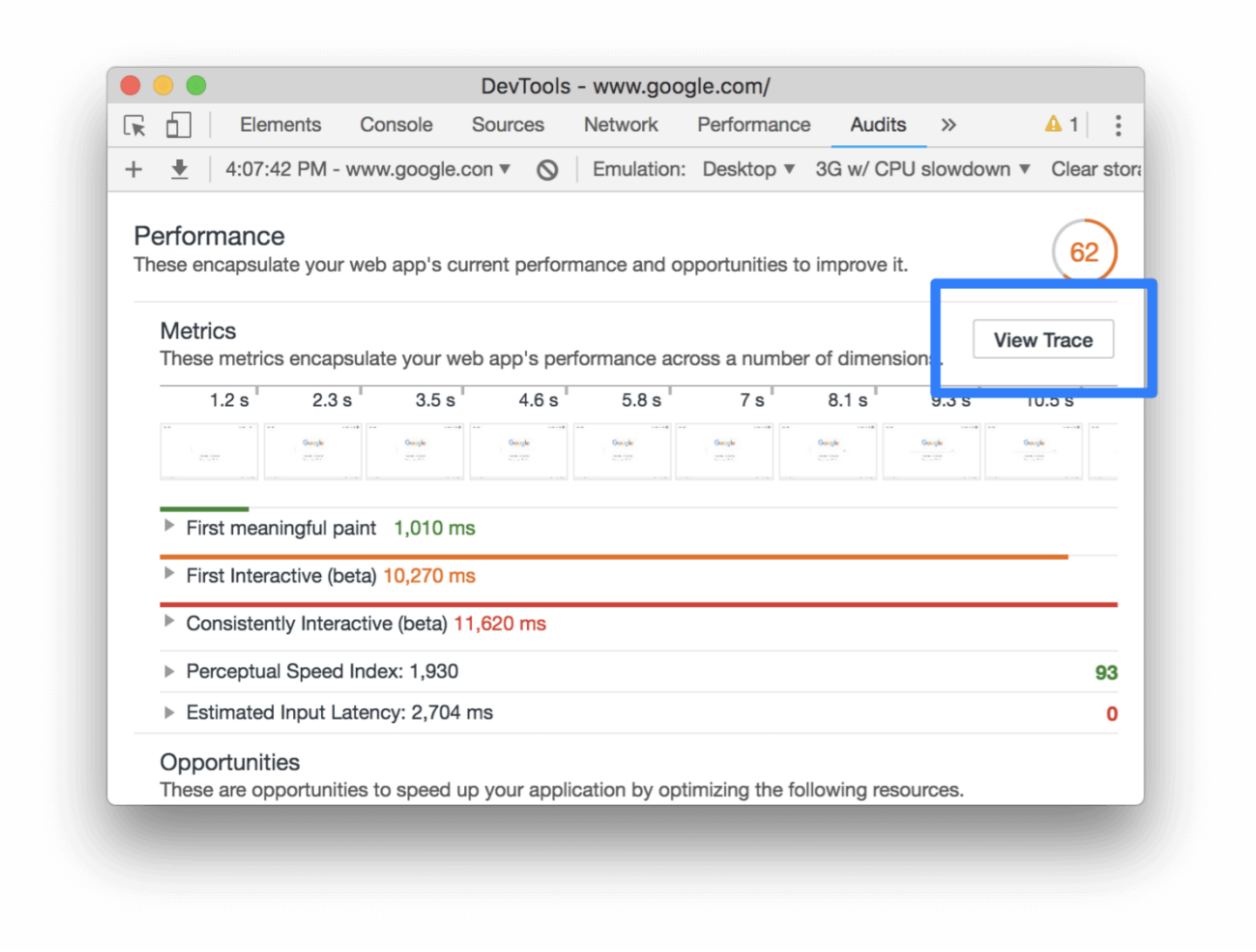

Figure 7. The **View Trace** button

Stop infinite loops

If you work with `for` loops, `do...while` loops, or recursion a lot, you've probably executed an infinite loop by mistake while developing your site. To stop the infinite loop, you can now:

1. Open the **Sources** panel.
2. Click **Pause** ❚❚. The button changes to **Resume Script Execution** ▶.
3. Hold **Resume Script Execution** ▶ then select **Stop Current JavaScript Call** ■.

In the video above, the clock is being updated via a `setInterval()` timer. Clicking **Start Infinite Loop** runs a `do...while` loop that never stops. The interval resumes because it wasn't running when **Stop Current JavaScript Call** ■ was selected.

User Timing in the Performance tabs

When viewing a Performance recording, click the **User Timing** section to view User Timing measures in the **Summary**, **Bottom-Up**, **Call Tree** and **Event Log** tabs.

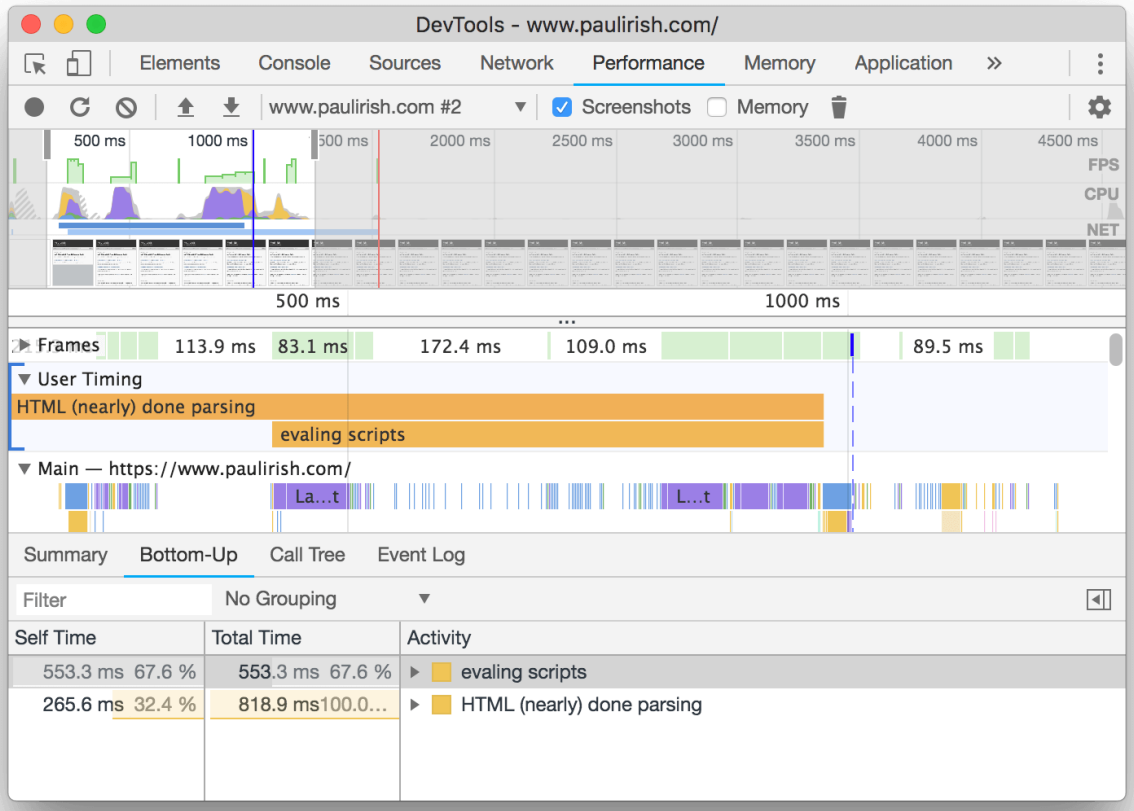

Figure 8. Viewing User Timing measures in the **Bottom-Up** tab. The blue bar to the left of the **User Timing** section indicates that it is selected.

In general, you can now select any of the sections (**Main Thread**, **User Timing**, **GPU**, **ScriptStreamer**, and so on) and view that section's activity in the tabs.

Select JavaScript VM instances in the Memory panel

The **Memory** panel now clearly lists out all JavaScript VM instances associated with a page, rather than hiding them behind the **Target** dropdown menu as before.

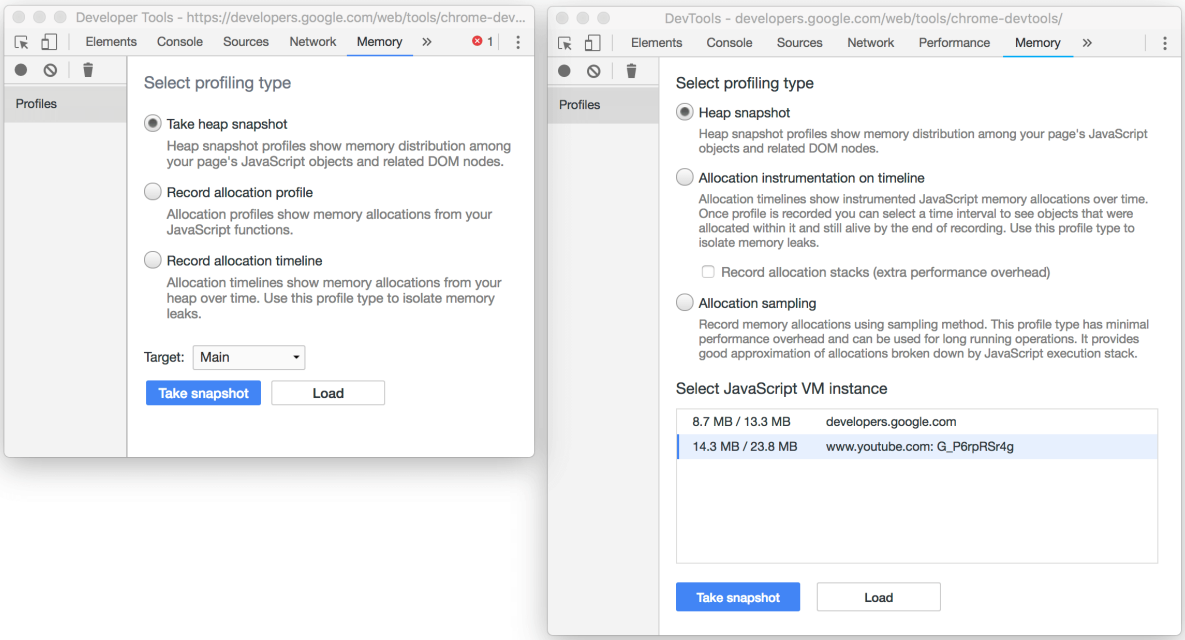

Figure 9. In the old UI on the left, the JavaScript VM instances are hidden behind the **Target** dropdown menu, whereas in the new UI on the right they are shown in the **Select JavaScript VM Instance** table

Next to the `developers.google.com` instance there are 2 values: `8.7 MB` and `13.3 MB`. The left value represents memory allocated because of JavaScript. The right value represents all OS memory that is being allocated because of that VM instance. The right value is inclusive of the left value. In Chrome's Task Manager, the left value corresponds to `JavaScript Memory` and the right value corresponds to `Memory Footprint`.

Network tab renamed to Page tab

On the **Sources** panel, the **Network** tab is now called the **Page** tab.

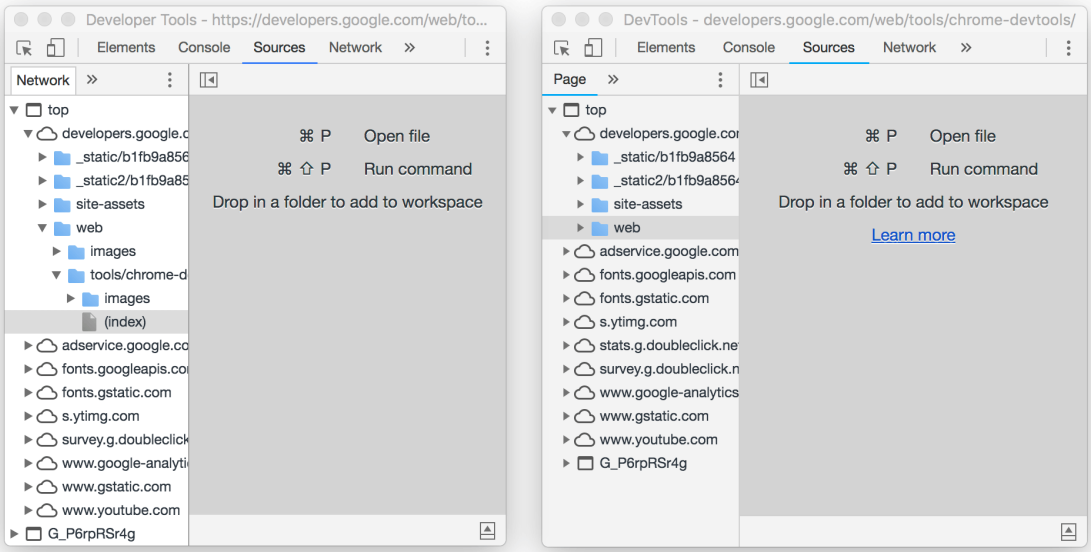

Figure 10. In the old UI on the left, the tab showing the page's resources is called **Network**, whereas in the new UI on the right it's called **Page**

Dark theme updates

Chrome 67 ships with a number of minor changes to the dark theme color scheme. For example, the breakpoint icons and the current line of execution are now green.

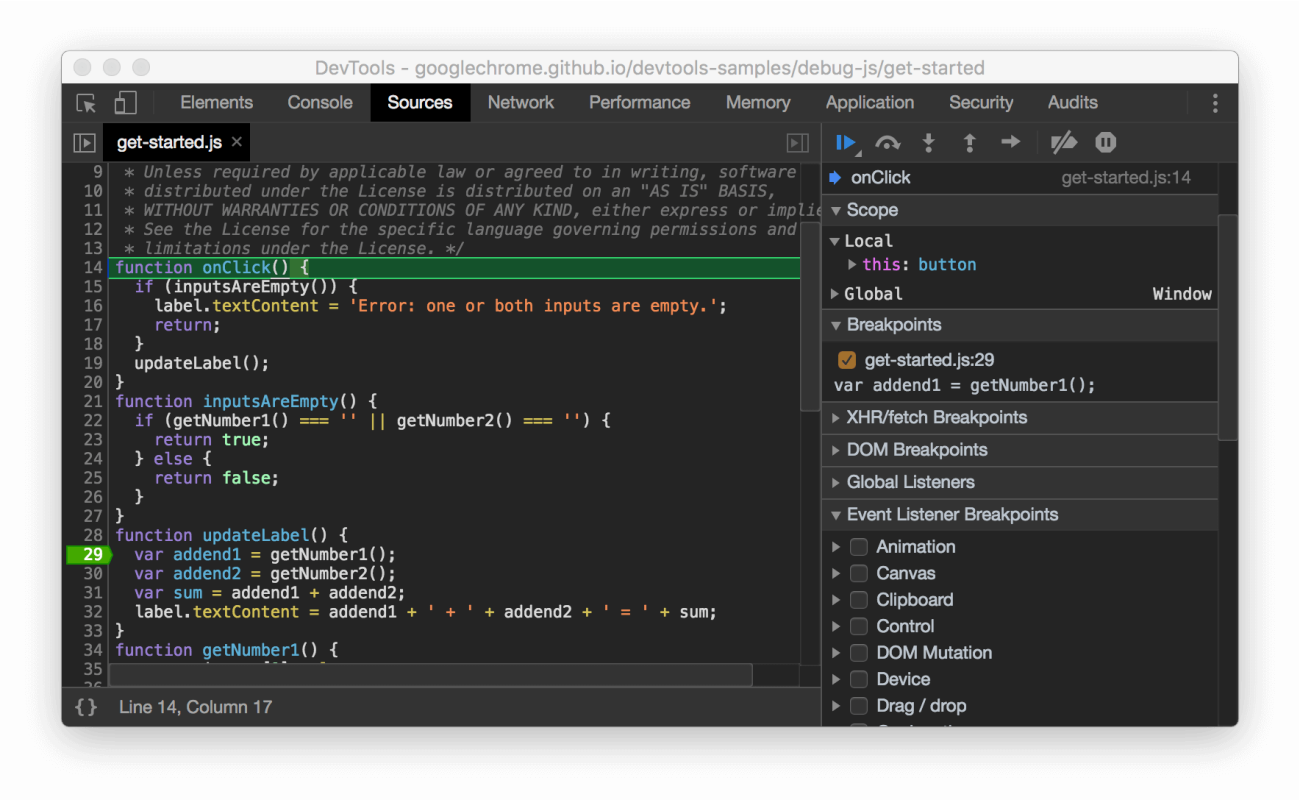

Figure 11. A screenshot of the new breakpoint icon and current line of execution color scheme

Certificate transparency in the Security panel

The **Security** panel now reports certificate transparency information.

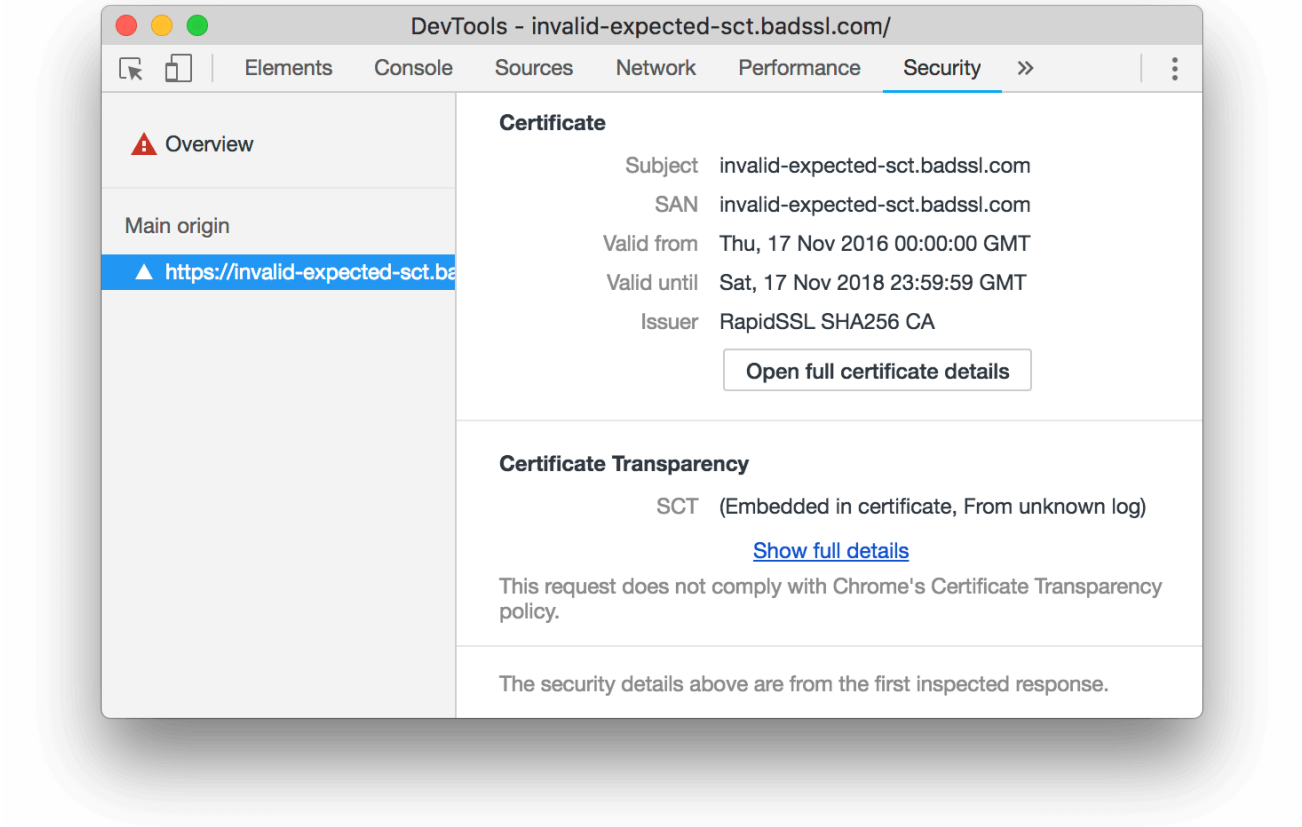

Figure 12. Certification transparency information in the Security panel

Site Isolation in the Performance panel

If you've got Site Isolation enabled, the **Performance** panel now provides a flame chart for each process so that you can see the total work that each process is causing.

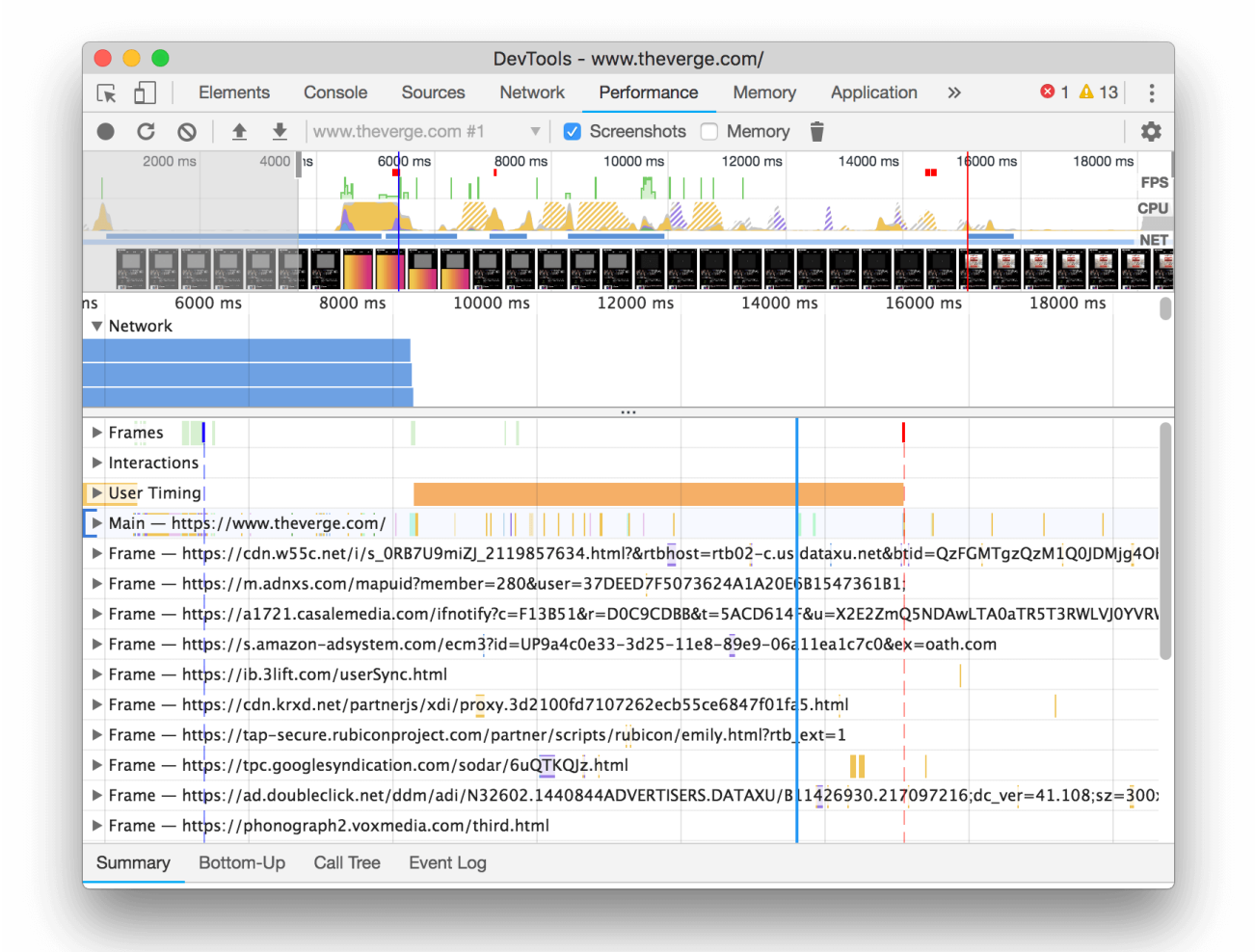

Figure 13. Per-process flame charts in a Performance recording

Feedback

That's all for Chrome 67!

Was this page helpful?

YES NO

To discuss the new features and changes in this post, or anything else related to DevTools:

- File bug reports at Chromium Bugs.
- Discuss features and changes on the Mailing List. Please don't use the mailing list for support questions. Use Stack Overflow, instead.

- Get help on how to use DevTools on Stack Overflow. Please don't file bugs on Stack Overflow. Use Chromium Bugs, instead.
- Tweet us at @ChromeDevTools.
- File bugs on this doc in the Web Fundamentals repository.

Consider Canary

If you're on Mac or Windows, please consider using Chrome Canary as your default development browser. If you report a bug or a change that you don't like while it's still in Canary, the DevTools team can address your feedback significantly faster.

Note: Canary is the bleeding-edge version of Chrome. It's released as soon as its built, without testing. This means that Canary breaks from time-to-time, about once-a-month, and it's usually fixed within a day. You can go back to using Chrome Stable while Canary is broken.

Previous release notes

See the devtools-whatsnew tag for links to all previous DevTools release notes.

Subscribe to our RSS or Atom feed and get the latest **updates** in your favorite feed reader!

Last updated July 2, 2018.