

# What's New In DevTools (Chrome 62)



By Kayce Basques

Technical Writer for Chrome DevTools

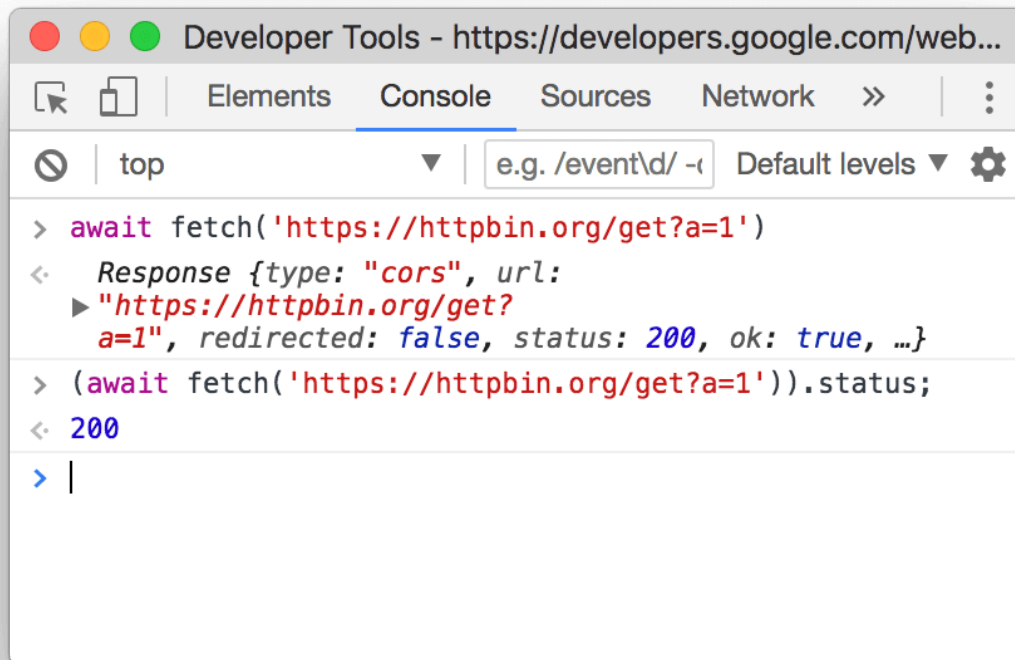
New features and changes coming to DevTools in Chrome 62:

- Support for top-level `await` operators in the **Console**.
- Screenshots of a portion of the viewport, and screenshots of specific HTML nodes.
- CSS Grid highlighting.
- A new **Console** API for querying objects.
- Negative filters and URL filters in the **Console**.
- HAR imports in the **Network** panel.
- Previewable cache resources.
- More predictable cache debugging.
- Block-level code coverage.

**Note:** You can check what version of Chrome you're running at <chrome://version>. Chrome auto-updates to a new major version about every 6 weeks.

## Top-level await operators in the Console

The **Console** now supports top-level await operators.




**Figure 1.** Using top-level await operators in the **Console**

## New screenshot workflows

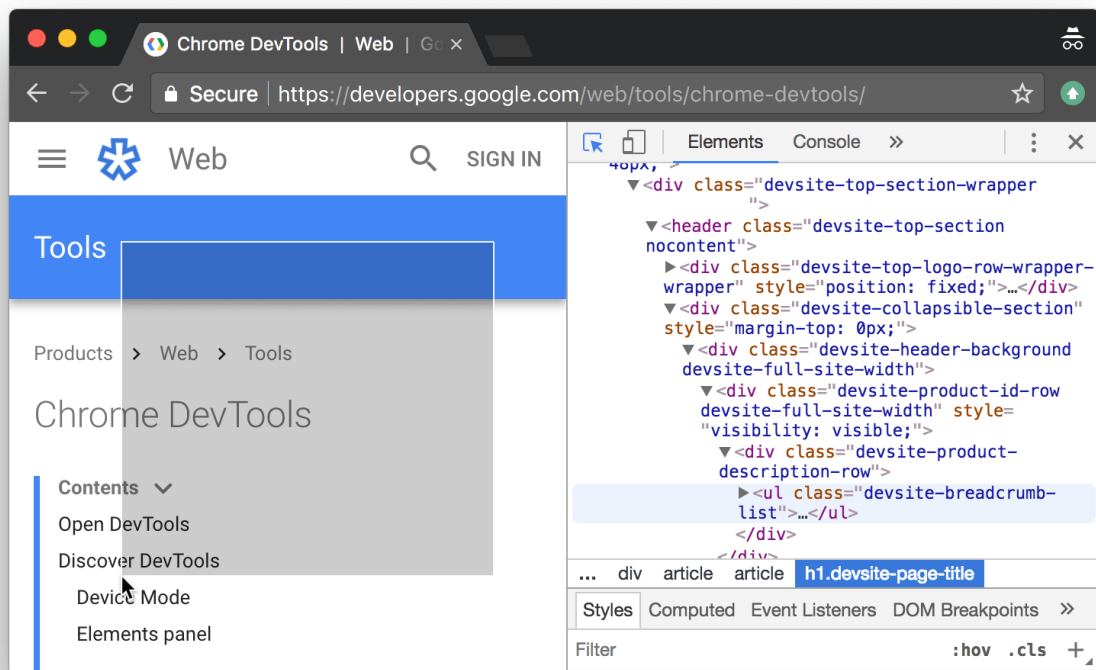
You can now take a screenshot of a portion of the viewport, or of a specific HTML node.

### Screenshots of a portion of the viewport

To take a screenshot of a portion of your viewport:

1. Click **Inspect**  or press Command+Shift+C (Mac) or Control+Shift+C (Windows, Linux) to enter Inspect Element Mode.
2. Hold Command (Mac) or Control (Windows, Linux) and select the portion of the viewport that you want to take a screenshot of.

3. Release your mouse. DevTools downloads a screenshot of the portion that you selected.

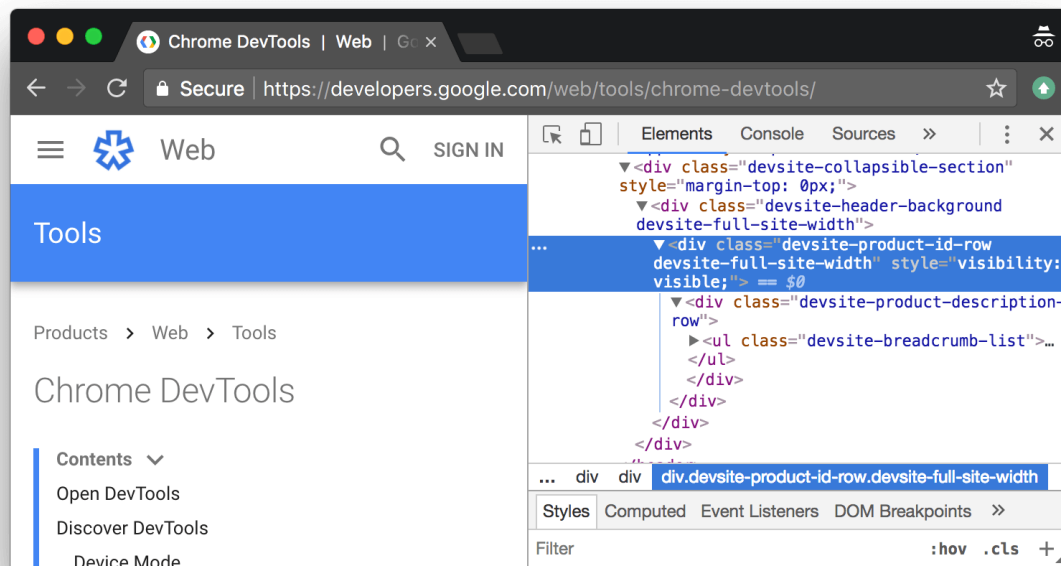


**Figure 2.** Taking a screenshot of a portion of the viewport

## Screenshots of specific HTML nodes

To take a screenshot of a specific HTML node:

1. Select an element in the **Elements** panel.



**Figure 3.** In this example, the goal is to take a screenshot of the blue header that contains the text **Tools**. Note that this node is already selected in the **DOM Tree** of the **Elements** panel

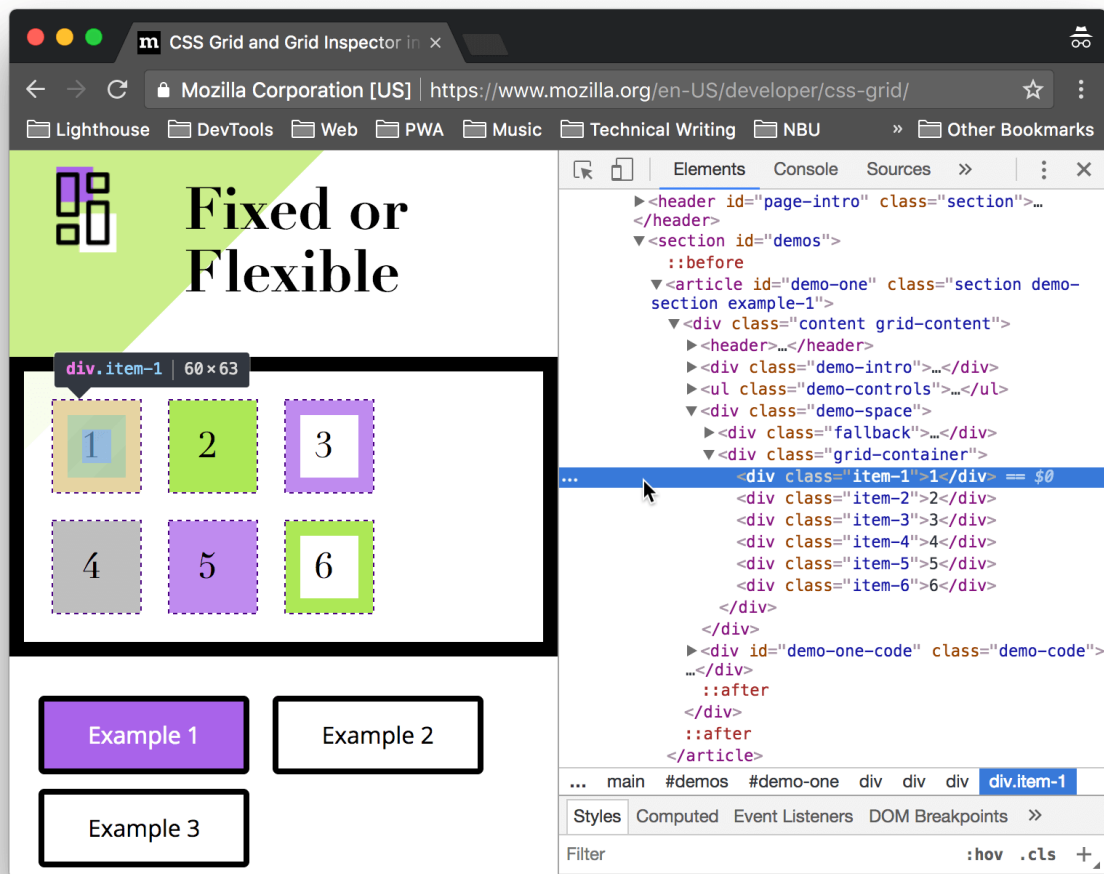
2. Open the Command Menu.
3. Start typing **node** and select **Capture node screenshot**. DevTools downloads a screenshot of the selected node.



**Figure 4.** The result of the **Capture node screenshot** command

## CSS Grid highlighting

To view the CSS Grid that's affecting an element, hover over an element in the **DOM Tree** of the **Elements** panel. A dashed border appears around each of the grid items. This only works when the selected item, or the parent of the selected item, has **display:grid** applied to it.



**Figure 5.** Highlighting a CSS Grid

Check out the video below to learn the basics of CSS Grid in less than 2 minutes.

## A new API for querying heap objects

Call `queryObjects(Constructor)` from the **Console** to return an array of objects that were created with the specified constructor. For example:

- `queryObjects(Promise)`. Returns all Promises.
- `queryObjects(HTMLElement)`. Returns all HTML elements.
- `queryObjects(foo)`, where `foo` is a function name. Returns all objects that were instantiated via `new foo()`.

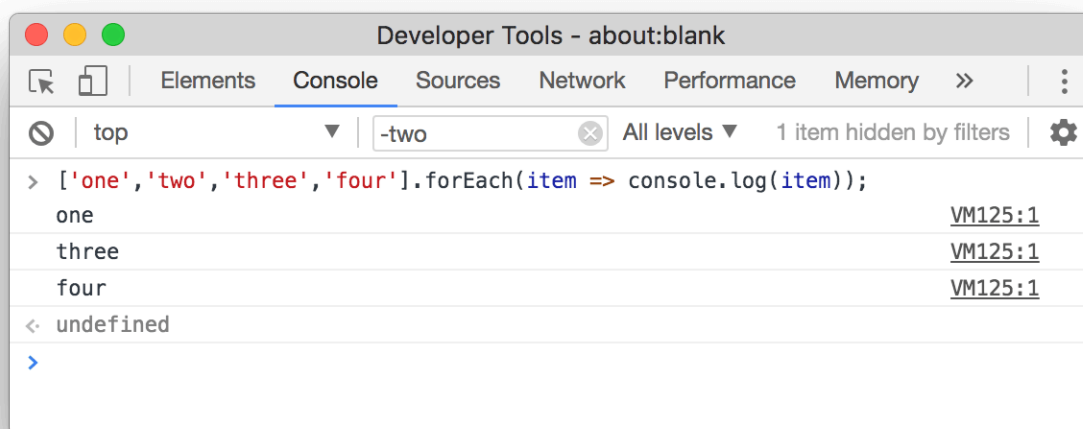
The scope of `queryObjects()` is the currently-selected execution context in the **Console**. See [Selecting execution context](#).

## New Console filters

The **Console** now supports negative and URL filters.

### Negative filters

Type `-<text>` in the **Filter** box to filter out any **Console** message that includes `<text>`.



**Figure 6.** The first statement logs one, two, three, and four to the **Console**. two is hidden because `-two` is entered in the **Filter** box

DevTools filters out a message if `<text>` is found:

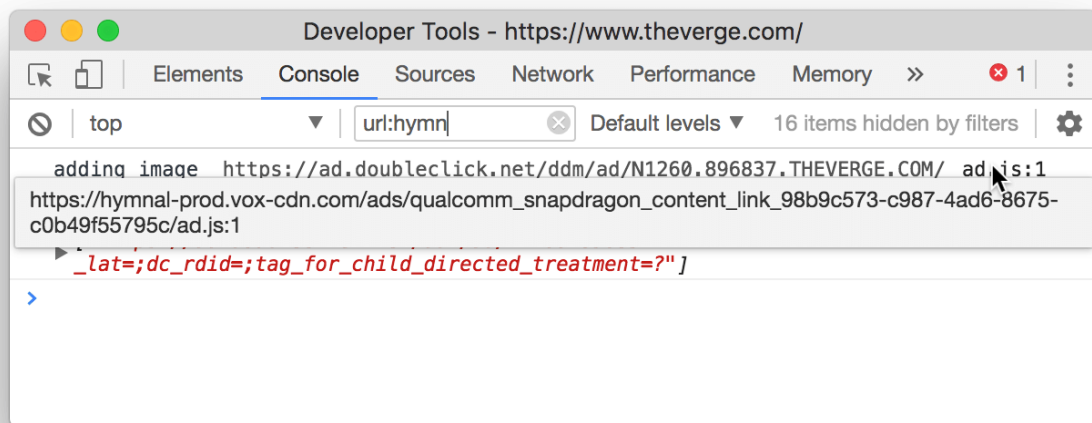
- In the message text.
- In the filename from which the message originated.
- In the stack trace text.

The negative filter also works with regular expressions such as `-/[4-5]*ms/`.

## URL filters

Type `url:<text>` in the **Filter** box to only show messages that originated from a script whose URL includes `<text>`.

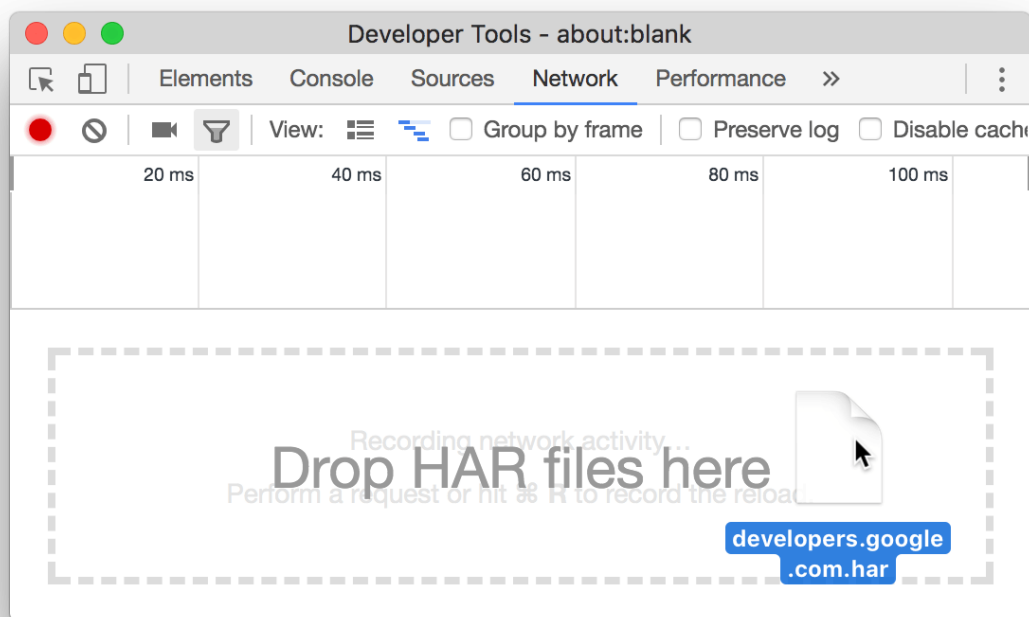
The filter uses fuzzy matching. If `<text>` appears anywhere in the URL, then DevTools shows the message.



**Figure 7.** Using URL filtering to only display messages that originate from scripts whose URL includes `hymn`. By hovering over the script name, you can see that the host name includes this text

## HAR imports in the Network panel

Drag and drop a HAR file into the **Network** panel to import it.



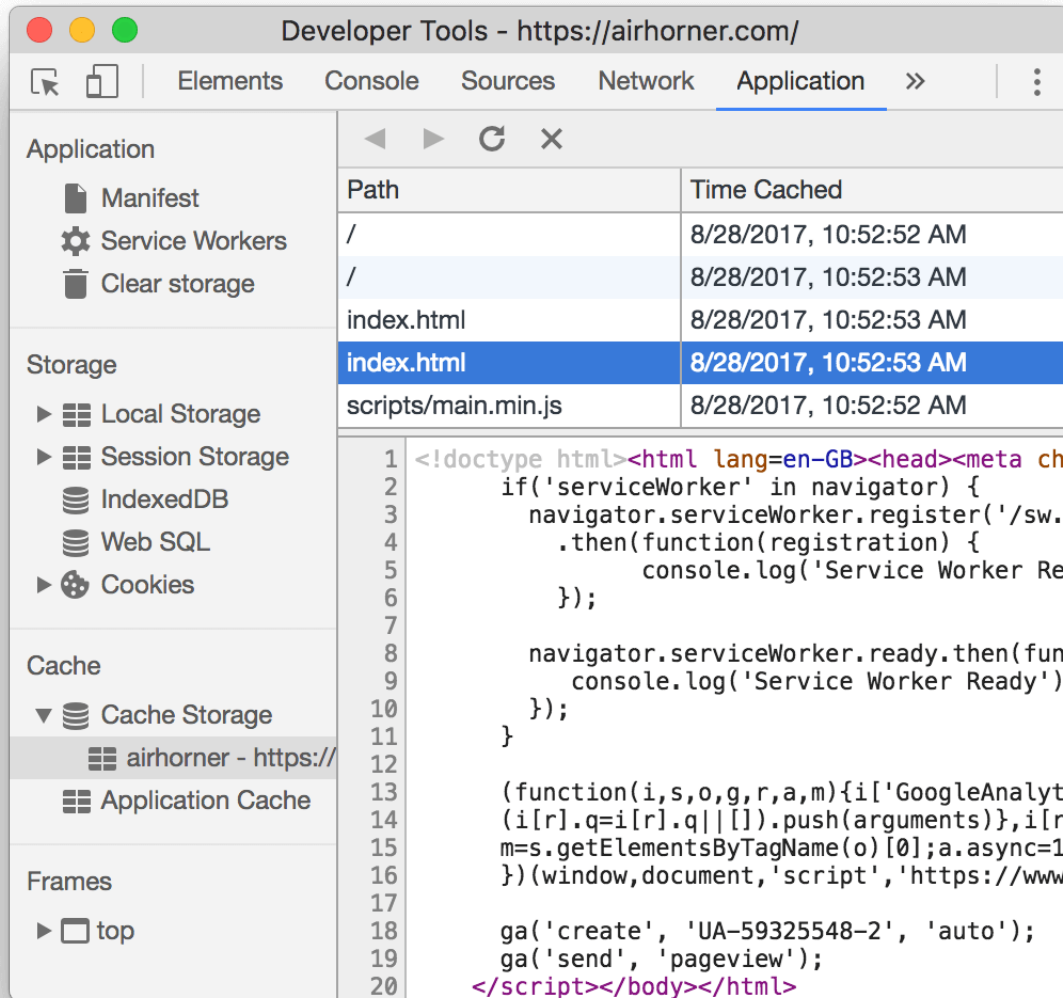
**Figure 8.** Importing a HAR file

**Note:** To export a HAR file, right-click a request and select **Save As HAR With Content**. All requests that DevTools has recorded are saved to the file. If you've got any filters enabled, those are ignored.

## Previewable cache resources in the Application panel

Click a row in a **Cache Storage** table to see a preview of that resource below the table.





**Figure 9.** Previewing a cache resource

## More responsive cache debugging

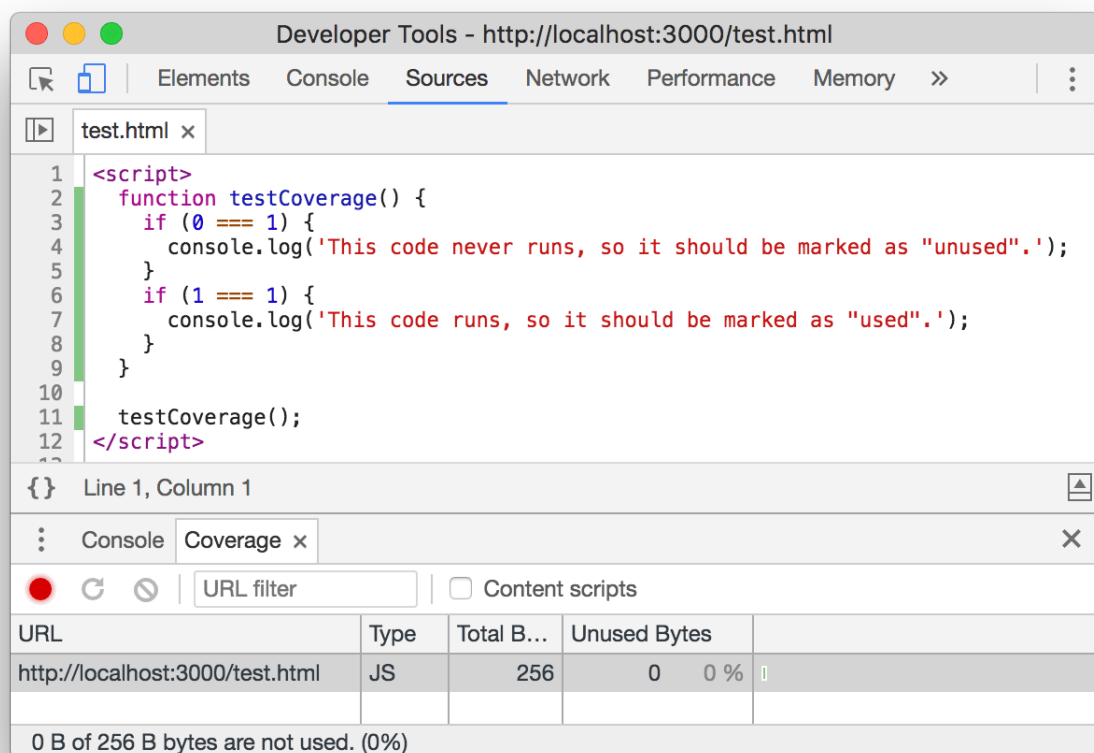
In Chrome 61 and earlier, debugging caches created with the [Cache API](#) is... rough. For example, when a page creates a new cache, you have to manually refresh the page or DevTools in order to see the new cache.

In Chrome 62, the **Cache Storage** tab now updates in real-time whenever you create, update, or delete a cache or a resource. Watch the video below for an example.

See the [Cache Storage Demo](#) to try it out yourself.

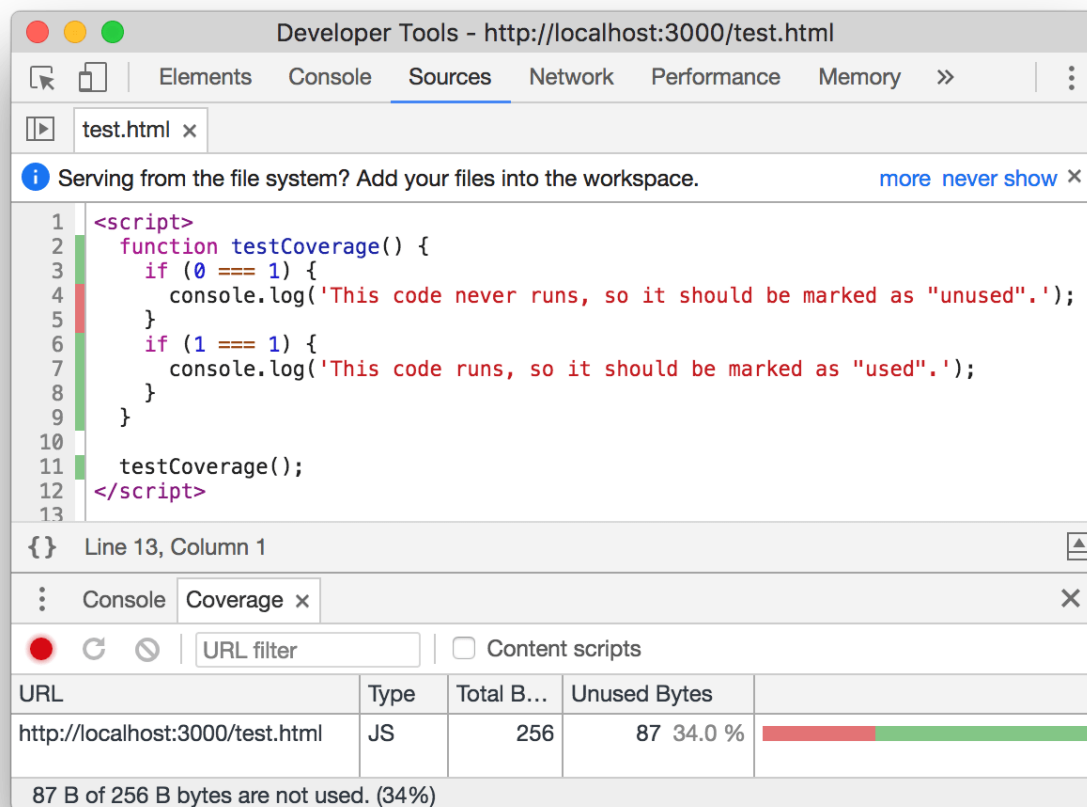
## Block-level code coverage

In Chrome 61 and earlier, the **Coverage** tab marks all of the code within a function as used, so long as the function is called.



**Figure 10.** An example of the **Coverage** tab in Chrome 61. Line 4 is marked used, even though it never executes

Starting in Chrome 62, the **Coverage** tab now tells you which code within a function is called.



**Figure 11.** An example of the **Coverage** tab in Chrome 62. Line 4 is marked unused

## Feedback

The best place to discuss any of the features or changes you see here is the [google-chrome-developer-tools@googlegroups.com mailing list](mailto:google-chrome-developer-tools@googlegroups.com). You can also tweet us at [@ChromeDevTools](https://twitter.com/ChromeDevTools) if you're short on time. If you've encountered a certain bug in DevTools, please [open an issue](#).

## Previous release notes

- [What's New In DevTools \(Chrome 61\)](#).
- [What's New In DevTools \(Chrome 60\)](#).
- [What's New In DevTools \(Chrome 59\)](#).
- [What's New In DevTools \(Chrome 58\)](#).

---

*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.*