

# Network Analysis Reference



By [Kayce Basques](#)

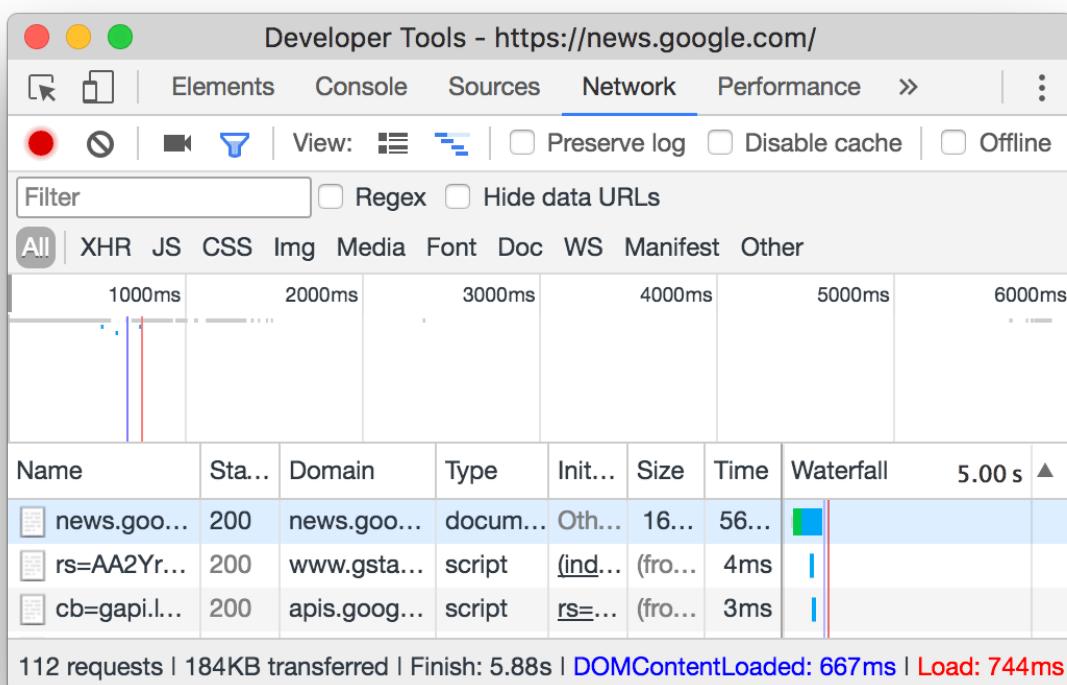
Technical Writer for Chrome DevTools

Discover new ways to analyze how your page loads in this comprehensive reference of Chrome DevTools network analysis features.

**Note:** This reference is based on Chrome 58. If you use another version of Chrome, the UI and features of DevTools may be different. Check <chrome://help> to see what version of Chrome you're running.

## Record network requests

By default, DevTools records all network requests in the Network panel, so long as DevTools is open.



**Figure 1.** The Network panel

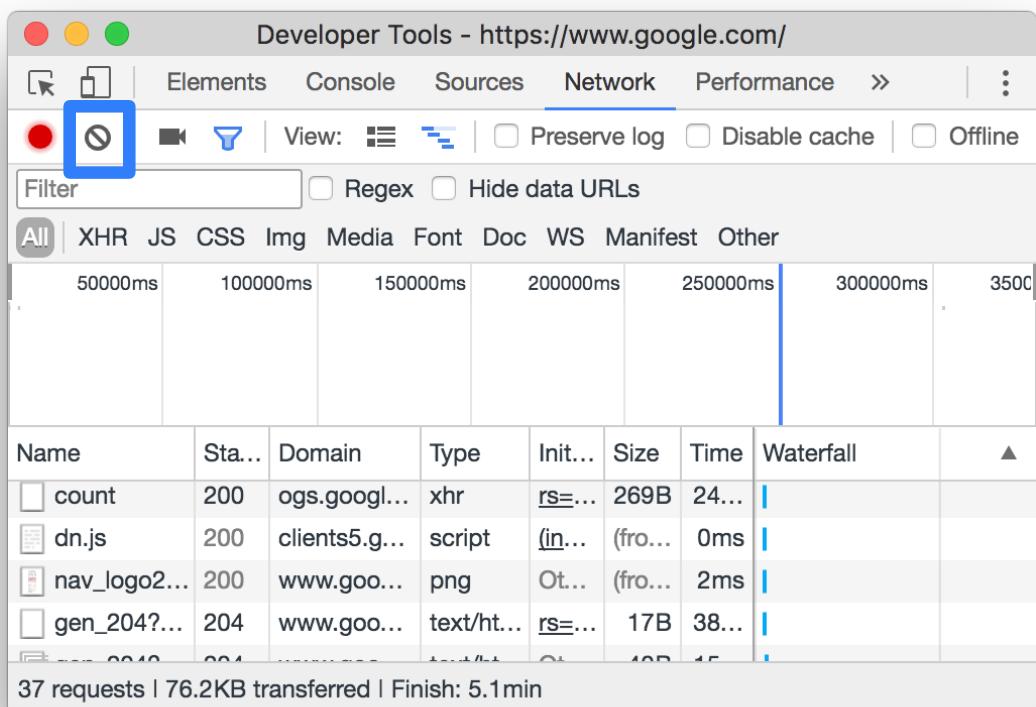
## Stop recording network requests

To stop recording requests:

- Click **Stop recording network log** (red circle) on the Network panel. It turns grey to indicate that DevTools is no longer recording requests.
- Press Command+E (Mac) or Control+E (Windows, Linux) while the Network panel is in focus.

## Clear requests

Click **Clear** (trash icon) on the Network panel to clear all requests from the Requests table.

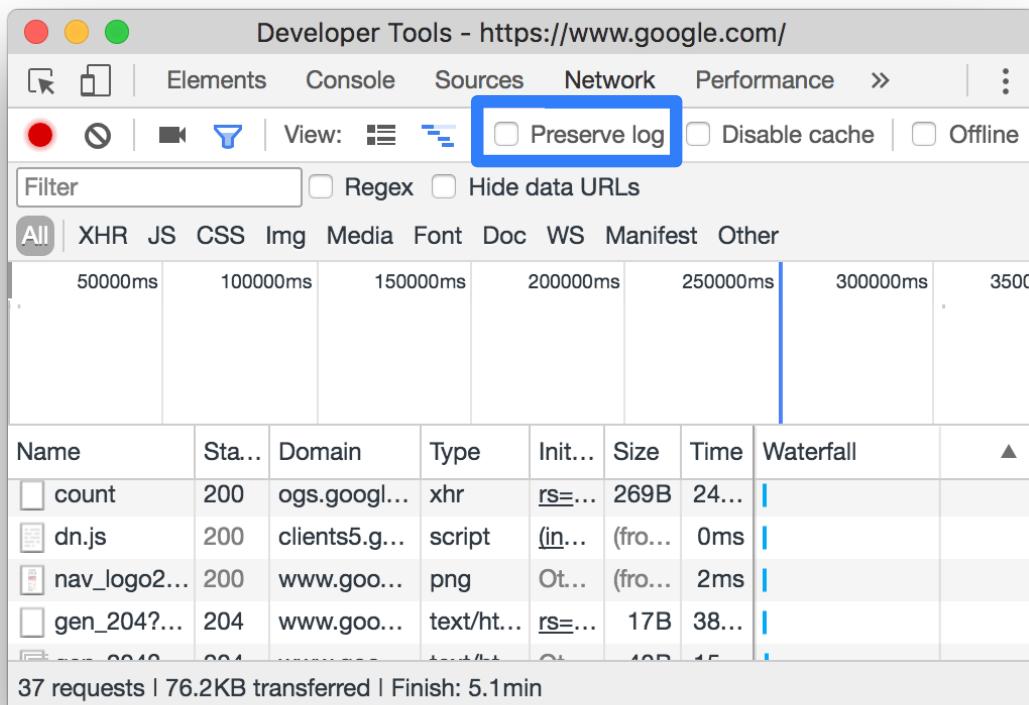


The screenshot shows the Network tab of the Developer Tools for the URL <https://www.google.com>. The 'Clear' button, which is a trash can icon, is highlighted with a blue border. The Requests table below lists several network requests, including 'count', 'dn.js', 'nav\_logo2...', 'gen\_204...', and '204'. The table includes columns for Name, Status, Domain, Type, Init..., Size, Time, and Waterfall. At the bottom of the table, it says '37 requests | 76.2KB transferred | Finish: 5.1min'.

**Figure 2.** Clear, outlined in blue

## Save requests across page loads

To save requests across page loads, check the **Preserve log** checkbox on the Network panel. DevTools saves all requests until you disable **Preserve log**.



**Figure 3.** The Preserve Log checkbox, outlined in blue

## Capture screenshots during page load

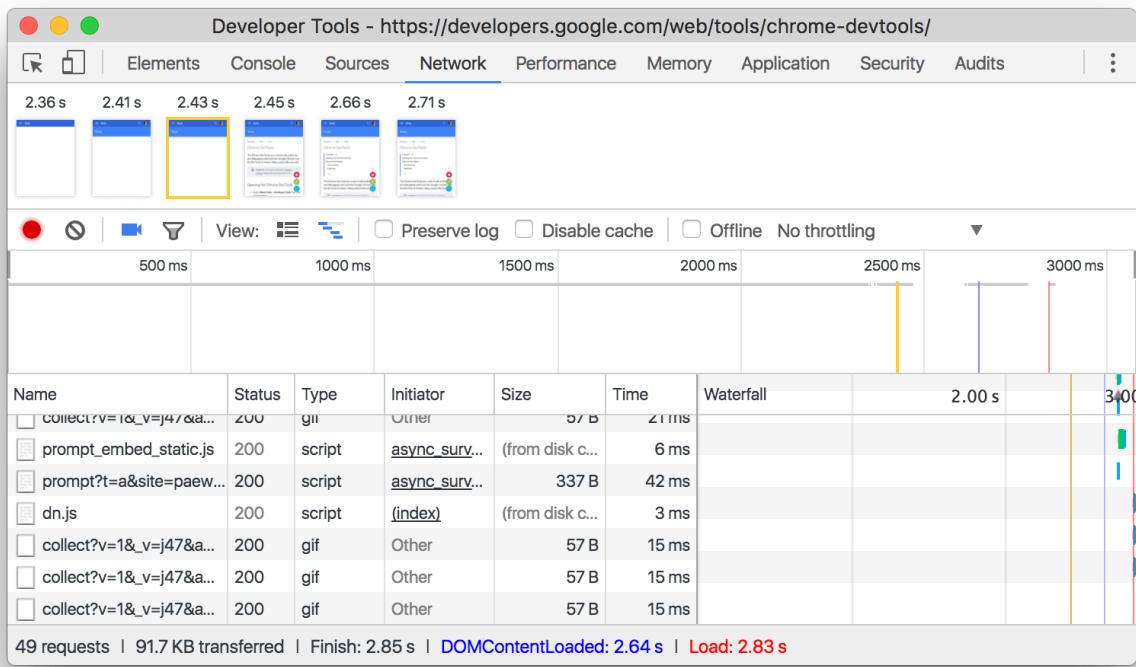
Capture screenshots to analyze what users see as they wait for your page to load.

To enable screenshots, click **Capture screenshots** on the Network panel. It turns blue when enabled.

Reload the page while the Network panel is in focus to capture screenshots.

Once captured, you can interact with screenshots in the following ways:

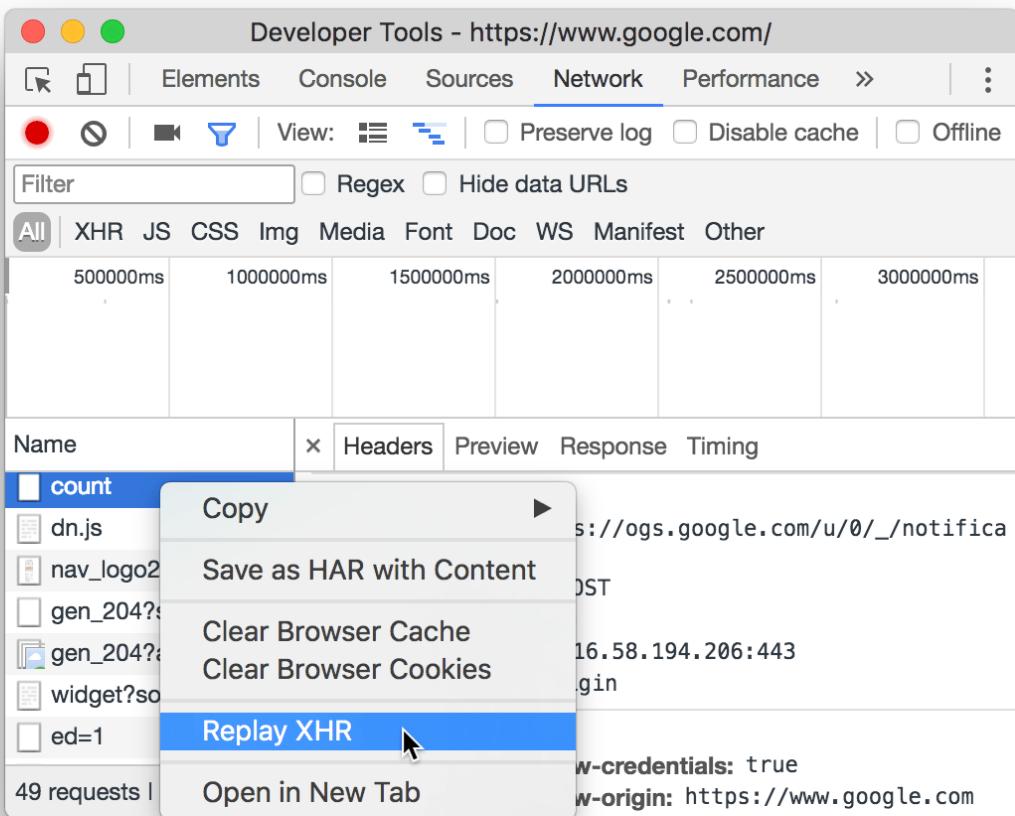
- Hover over a screenshot to view the point at which that screenshot was captured. A yellow line appears on the Overview pane.
- Click a screenshot's thumbnail to filter out any requests that occurred after the screenshot was captured.
- Double-click a thumbnail to zoom in on it.



**Figure 4.** Hovering over a screenshot. The yellow, vertical line in the Overview pane and the Waterfall represent the time at which the screenshot was captured.

## Replay XHR request

To replay an XHR request, right-click the request in the Requests table and select **Replay XHR**.

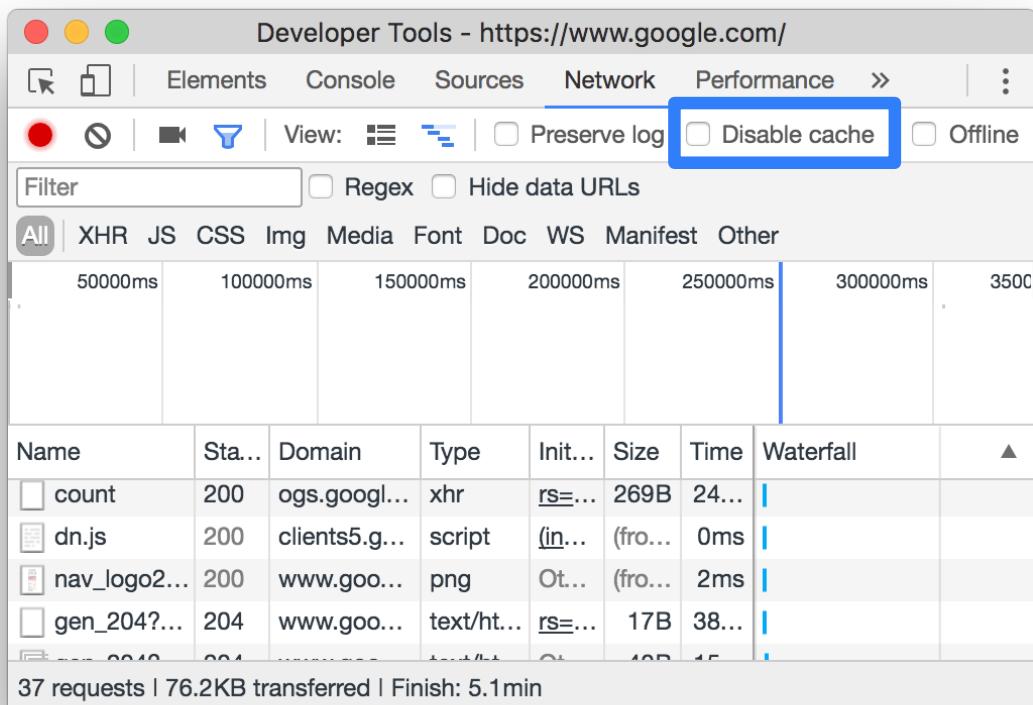


**Figure 5.** Selecting Replay XHR

## Change loading behavior

### Emulate a first-time visitor by disabling the browser cache

To emulate how a first-time user experiences your site, check the **Disable cache** checkbox. DevTools disables the browser cache. This more accurately emulates a first-time user's experience, because requests are served from the browser cache on repeat visits.



**Figure 6.** The Disable Cache checkbox, outlined in blue

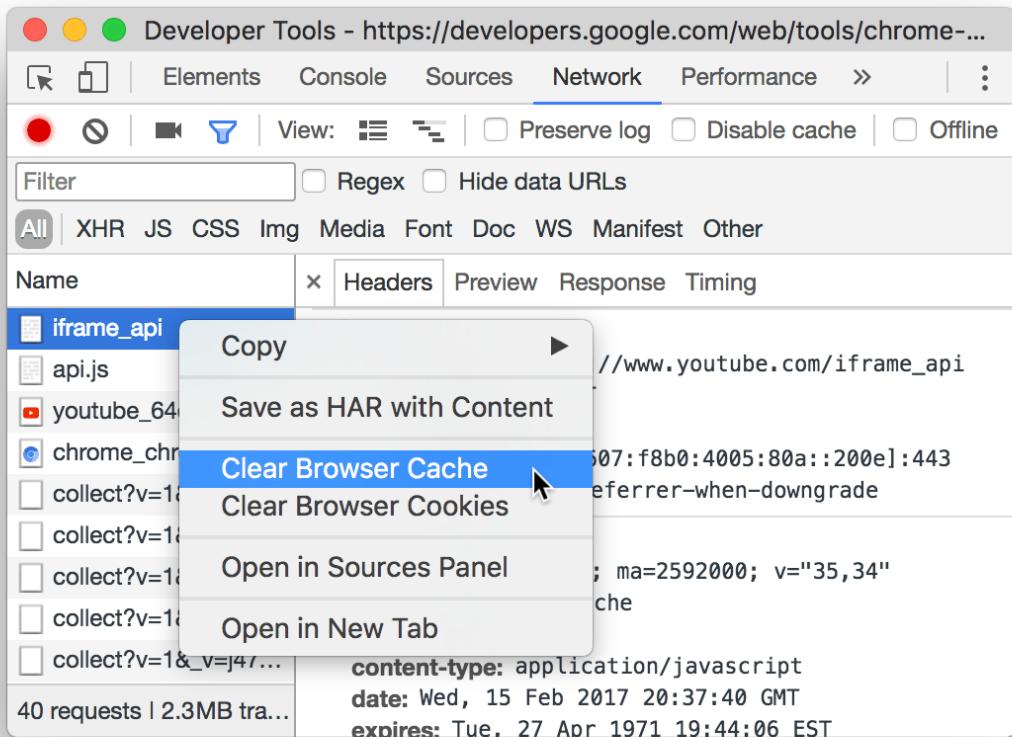
### Disable the browser cache from the Network Conditions drawer

If you want to disable the cache while working in other DevTools panels, use the Network Conditions drawer.

1. Open the Network Conditions drawer.
2. Check or uncheck the **Disable cache** checkbox.

### Manually clear the browser cache

To manually clear the browser cache at any time, right-click anywhere in the Requests table and select **Clear Browser Cache**.

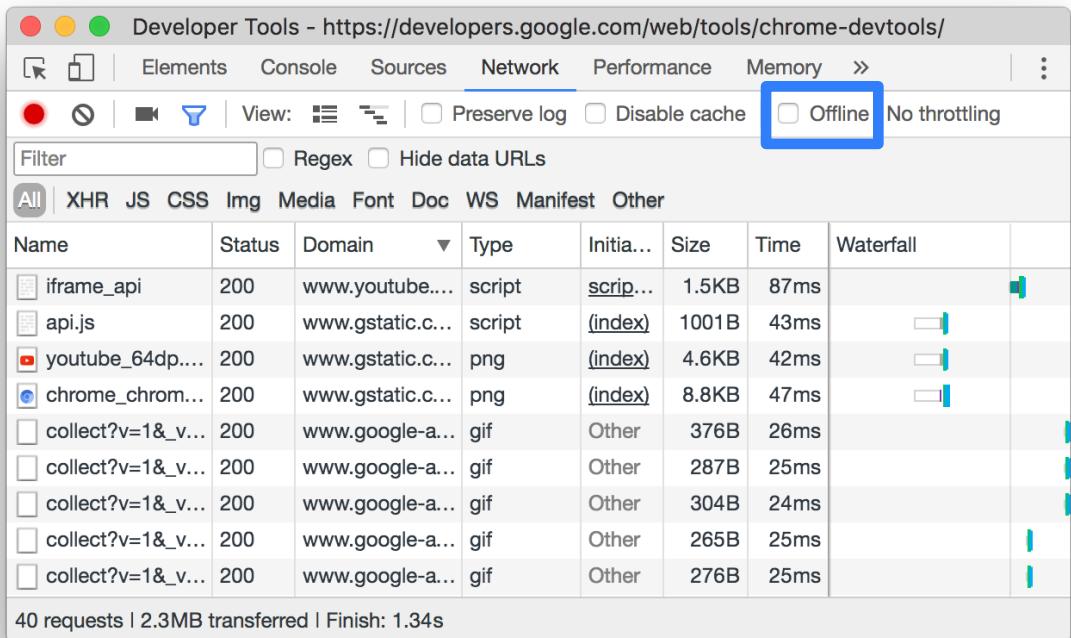


**Figure 7.** Selecting Clear Browser Cache

## Emulate offline

There's a new class of web apps, called [Progressive Web Apps](#), which can function offline with the help of [service workers](#). When you're building this type of app, it's useful to be able to quickly simulate a device that has no data connection.

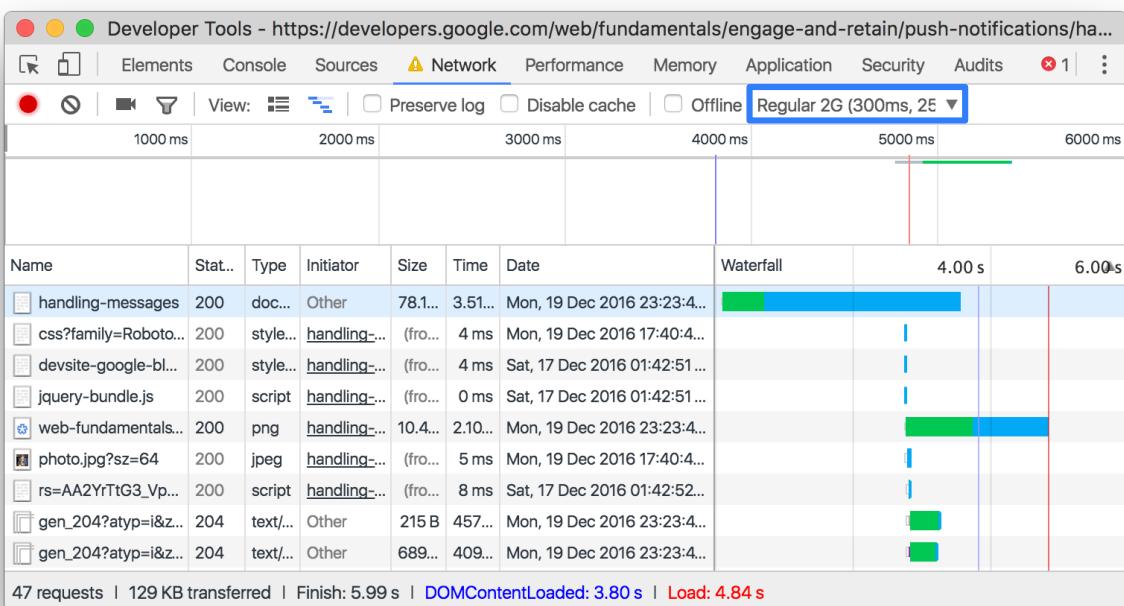
Check the **Offline** checkbox to simulate a completely offline network experience.



**Figure 8.** The Offline checkbox, outlined in blue

## Emulate slow network connections

Emulate 2G, 3G, and other connection speeds from the **Network Throttling** menu.



### **Figure 9.** The Network Throttling menu, outlined in blue

You can select from a variety of presets, such as Regular or Good 2G. You can also add your own custom presets by opening the Network Throttling menu and selecting **Custom > Add**.

DevTools displays a warning icon next to the **Network** tab to remind you that throttling is enabled.

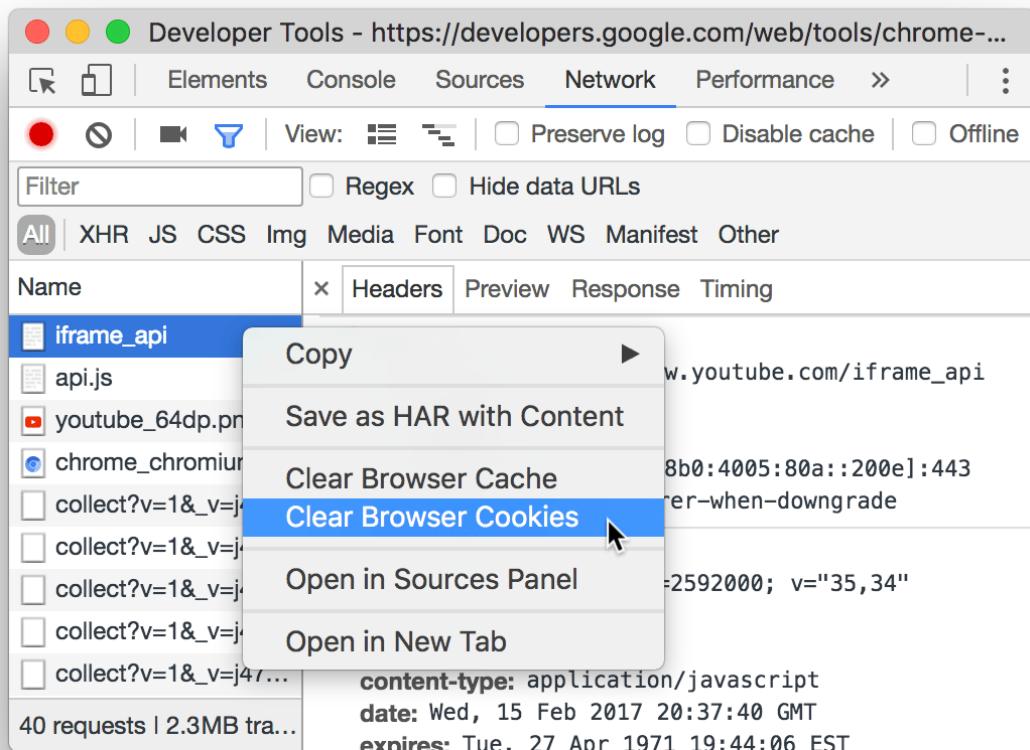
### **Emulate slow network connections from the Network Conditions drawer**

If you want to throttle the network connection while working in other DevTools panels, use the Network Conditions drawer.

1. Open the Network Conditions drawer.
2. Select your desired connection speed from the **Network Throttling** menu.

### **Manually clear browser cookies**

To manually clear browser cookies at any time, right-click anywhere in the Requests table and select **Clear Browser Cookies**.



**Figure 10.** Selecting Clear Browser Cookies

## Override the user agent

To manually override the user agent:

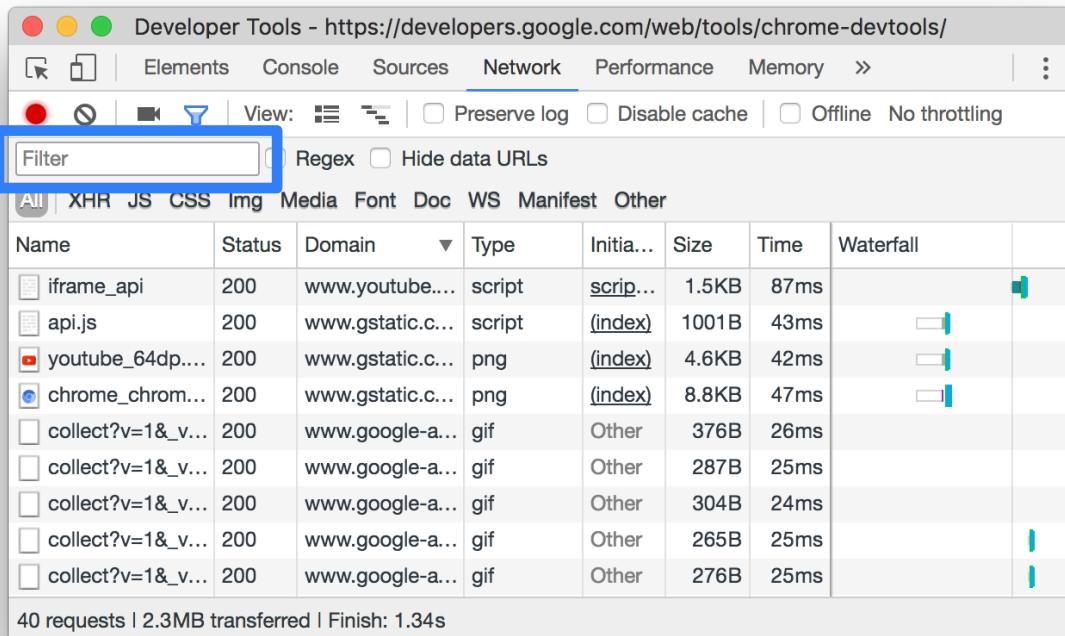
1. Open the [Network Conditions drawer](#).
2. Uncheck **Select automatically**.
3. Choose a user agent option from the menu, or enter a custom one in the text box.

## Filter requests

### Filter requests by properties

Use the **Filter** text box to filter requests by properties, such as the domain or size of the request.

If you can't see the text box, the Filters pane is probably hidden. See [Hide the Filters pane](#).



**Figure 11.** The Filters text box, outlined in blue

You can use multiple properties simultaneously by separating each property with a space. For example, `mime-type:image/gif larger-than:1K` displays all GIFs that are larger than one kilobyte. These multi-property filters are equivalent to AND operations. OR operations are currently not supported.

Below is a complete list of supported properties.

- **domain.** Only display resources from the specified domain. You can use a wildcard character (\*) to include multiple domains. For example, `*.com` displays resources from all domain names ending in `.com`. DevTools populates the autocomplete dropdown menu with all of the domains it has encountered.
- **has-response-header.** Show the resources that contain the specified HTTP response header. DevTools populates the autocomplete dropdown with all of the response headers that it has encountered.
- **is.** Use `is:running` to find WebSocket resources.
- **larger-than.** Show resources that are larger than the specified size, in bytes. Setting a value of `1000` is equivalent to setting a value of `1k`.
- **method.** Show resources that were retrieved over a specified HTTP method type. DevTools populates the dropdown with all of the HTTP methods it has encountered.

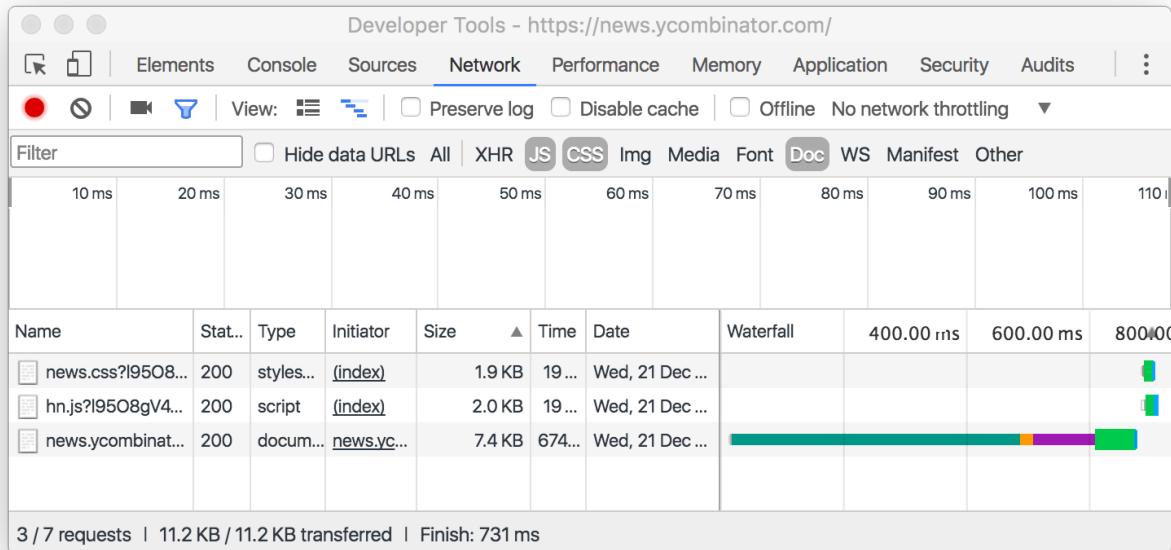
- **mime-type**. Show resources of a specified MIME type. DevTools populates the dropdown with all MIME types it has encountered.
- **mixed-content**. Show all mixed content resources (`mixed-content:all`) or just the ones that are currently displayed (`mixed-content:displayed`).
- **scheme**. Show resources retrieved over unprotected HTTP (`scheme:http`) or protected HTTPS (`scheme:https`).
- **set-cookie-domain**. Show the resources that have a `Set-Cookie` header with a `Domain` attribute that matches the specified value. DevTools populates the autocomplete with all of the cookie domains that it has encountered.
- **set-cookie-name**. Show the resources that have a `Set-Cookie` header with a name that matches the specified value. DevTools populates the autocomplete with all of the cookie names that it has encountered.
- **set-cookie-value**. Show the resources that have a `Set-Cookie` header with a value that matches the specified value. DevTools populates the autocomplete with all of the cookie values that it has encountered.
- **status-code**. Only show resources whose HTTP status code match the specified code. DevTools populates the autocomplete dropdown menu with all of the status codes it has encountered.

## Filter requests by type

To filter requests by request type, click the **XHR**, **JS**, **CSS**, **Img**, **Media**, **Font**, **Doc**, **WS** (WebSocket), **Manifest**, or **Other** (any other type not listed here) buttons on the Network panel.

If you can't see these buttons, the Filters pane is probably hidden. See [Hide the Filters pane](#).

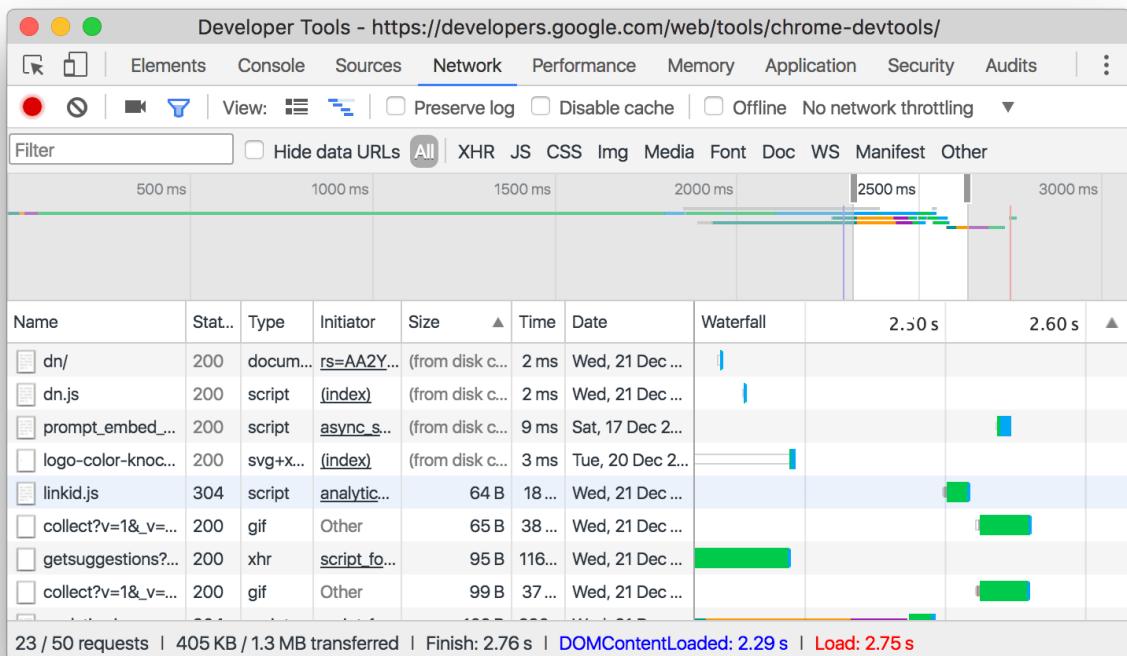
To enable multiple type filters simultaneously, hold Command (Mac) or Control (Windows, Linux) and then click.



**Figure 12.** Using the Type filters to display JS, CSS, and Doc[ument] resources.

## Filter requests by time

Click and drag left or right on the Overview pane to only display requests that were active during that time frame. The filter is inclusive. Any request that was active during the highlighted time is shown.



**Figure 13.** Filtering out any requests that weren't active around 2500ms

## Hide data URLs

Data URLs are small files embedded into other documents. Any request that you see in the Requests table that starts with `data:` is a data URL.

Check the **Hide data URLs** checkbox to hide these requests.

The screenshot shows the Chrome DevTools Network tab for the URL `https://developers.google.com/web/tools/chrome-devtools/`. The 'Network' tab is selected. At the top, there is a 'Filter' input field, a 'Regex' checkbox, and a 'Hide data URLs' checkbox, which is highlighted with a blue box. Below the toolbar, there are buttons for View (grid, list, timeline), Preserve log, Disable cache, Offline, and No throttling. The main area displays a table of network requests with the following columns: Name, Status, Domain, Type, Initia..., Size, Time, and Waterfall. The requests listed are: sources.png, console.png, elements.png, device-mode..., web-fundame..., jquery-bundle.js, devsite-googl..., and script\_foot.js. At the bottom of the table, it says '40 requests | 2.3MB transferred | Finish: 1.34s'.

**Figure 14.** The Hide Data URLs checkbox

## Sort requests

By default, the requests in the Requests table are sorted by initiation time, but you can sort the table using other criteria.

### Sort by column

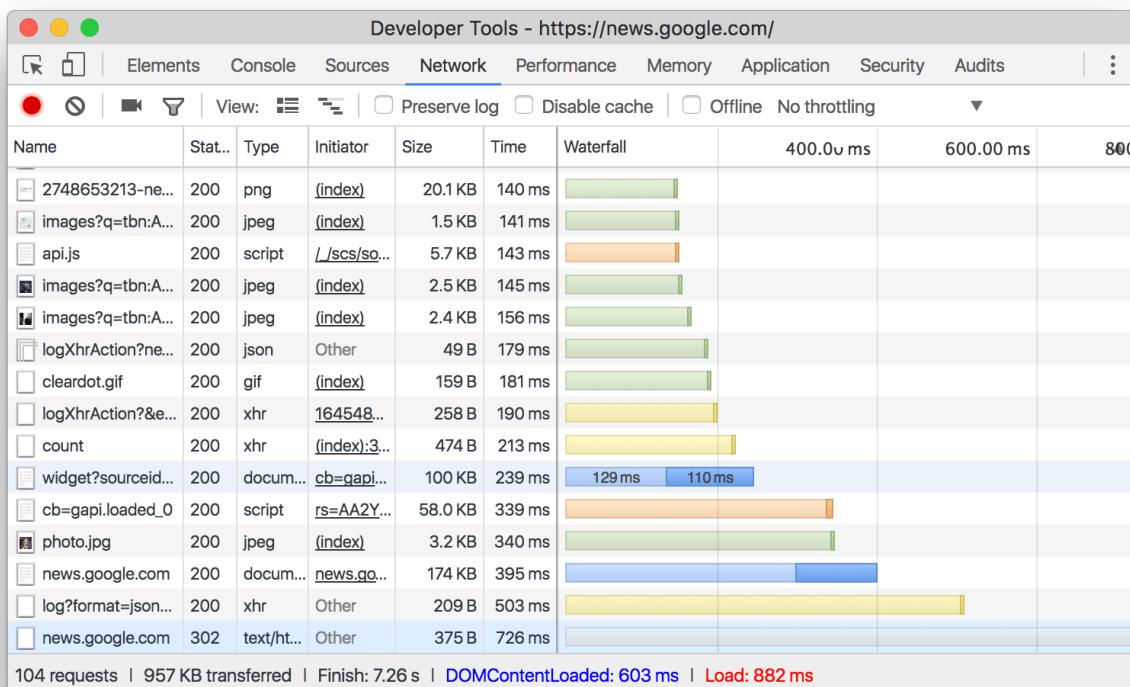
Click the header of any column in the Requests to sort requests by that column.

### Sort by activity phase

To change how the Waterfall sorts requests, right-click the header of the Requests table, hover over **Waterfall**, and select one of the following options:

- **Start Time.** The first request that was initiated is at the top.
- **Response Time.** The first request that started downloading is at the top.
- **End Time.** The first request that finished is at the top.
- **Total Duration.** The request with the shortest connection setup and request / response is at the top.
- **Latency.** The request that waited the shortest time for a response is at the top.

These descriptions assume that each respective option is ranked from shortest to longest. Clicking on the **Waterfall** column's header reverses the order.



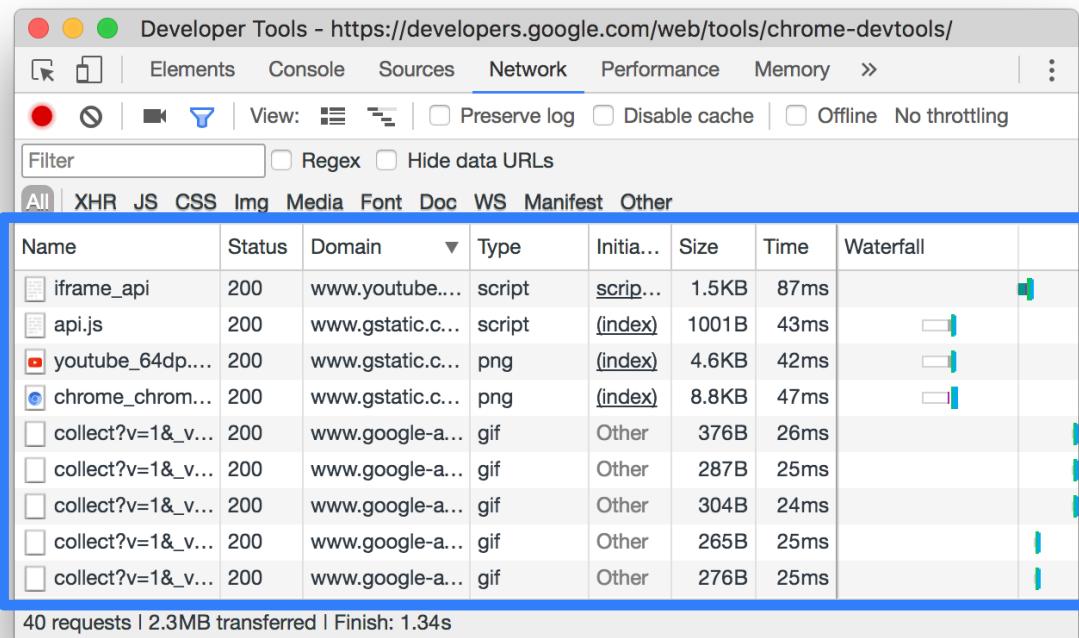
**Figure 15.** Sorting the Waterfall by total duration. The lighter portion of each bar is time spent waiting. The darker portion is time spent downloading bytes.

## Analyze requests

So long as DevTools is open, it logs all requests in the Network panel. Use the Network panel to analyze requests.

## View a log of requests

Use the Requests table to view a log of all requests made while DevTools has been open. Clicking or hovering over requests reveals more information about them.



The screenshot shows the Network tab in the Chrome DevTools developer tools interface. The table displays a list of network requests. The columns are: Name, Status, Domain, Type, Initiator, Size, Time, and Waterfall. The Waterfall column contains small waterfall charts for each request. The requests listed include various files like 'iframe\_api', 'api.js', and multiple 'collect' requests from 'www.google-analytics.com'. The table has a blue border around it.

Name	Status	Domain	Type	Initiator	Size	Time	Waterfall
iframe_api	200	www.youtube....	script	script...	1.5KB	87ms	
api.js	200	www.gstatic.c...	script	(index)	1001B	43ms	
youtube_64dp...	200	www.gstatic.c...	png	(index)	4.6KB	42ms	
chrome_chrom...	200	www.gstatic.c...	png	(index)	8.8KB	47ms	
collect?v=1&_v...	200	www.google-a...	gif	Other	376B	26ms	
collect?v=1&_v...	200	www.google-a...	gif	Other	287B	25ms	
collect?v=1&_v...	200	www.google-a...	gif	Other	304B	24ms	
collect?v=1&_v...	200	www.google-a...	gif	Other	265B	25ms	
collect?v=1&_v...	200	www.google-a...	gif	Other	276B	25ms	

**Figure 16.** The Requests table, outlined in blue

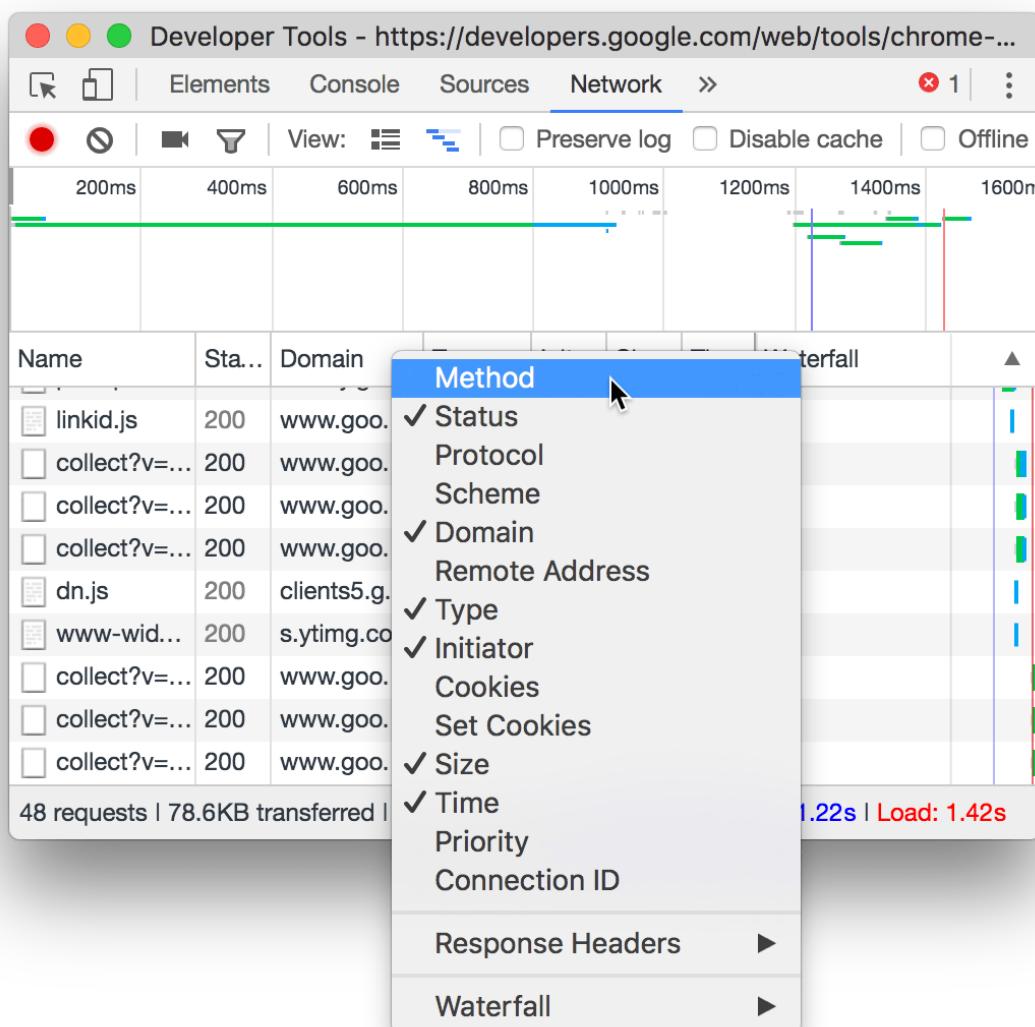
The Requests table displays the following columns by default:

- **Name.** The filename of, or an identifier for, the resource.
- **Status.** The HTTP status code.
- **Type.** The MIME type of the requested resource.
- **Initiator.** The following objects or processes can initiate requests:
  - **Parser.** Chrome's HTML parser.
  - **Redirect.** An HTTP redirect.
  - **Script.** A JavaScript function.
  - **Other.** Some other process or action, such as navigating to a page via a link or entering a URL in the address bar.
- **Size.** The combined size of the response headers plus the response body, as delivered by the server.

- **Time.** The total duration, from the start of the request to the receipt of the final byte in the response.
- **Waterfall.** A visual breakdown of each request's activity.

### Add or remove columns

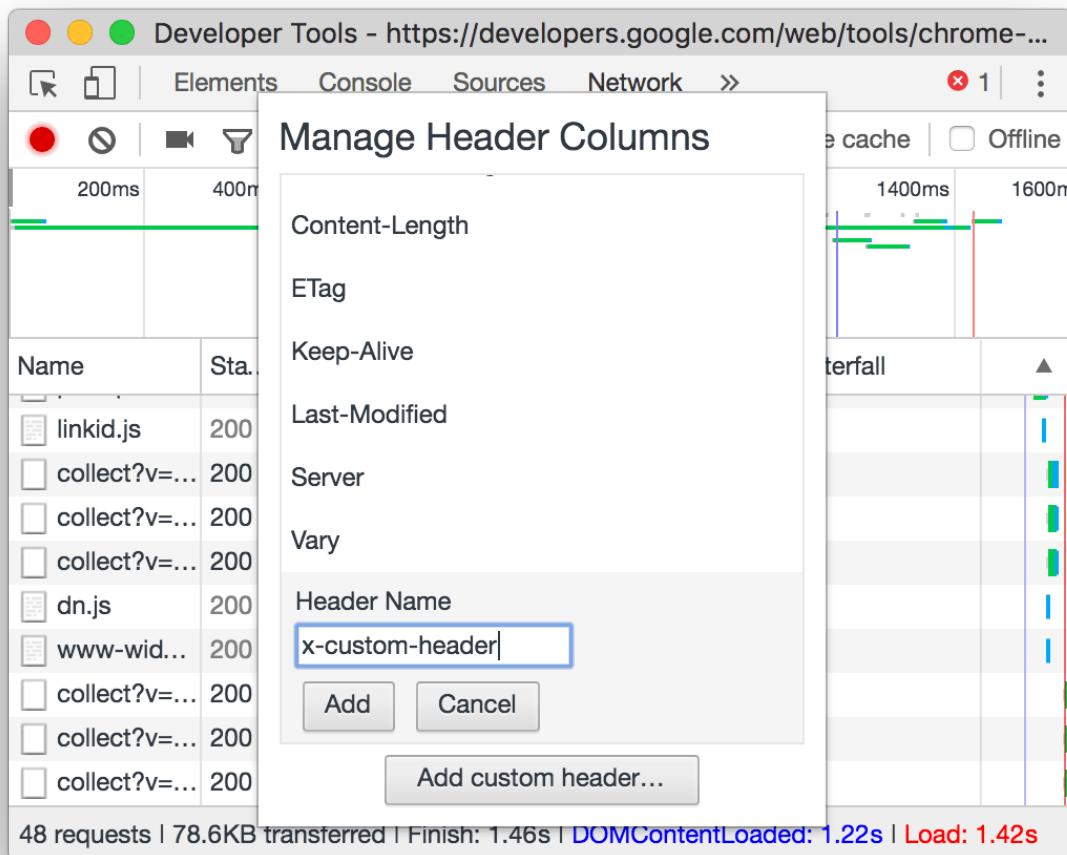
Right-click the header of the Requests table and select an option to hide or show it. Currently displayed options have checkmarks next to them.



**Figure 17.** Adding a column to the Requests table.

### Add custom columns

To add a custom column to the Requests table, right-click the header of the Requests table and select **Response Headers > Manage Header Columns**.

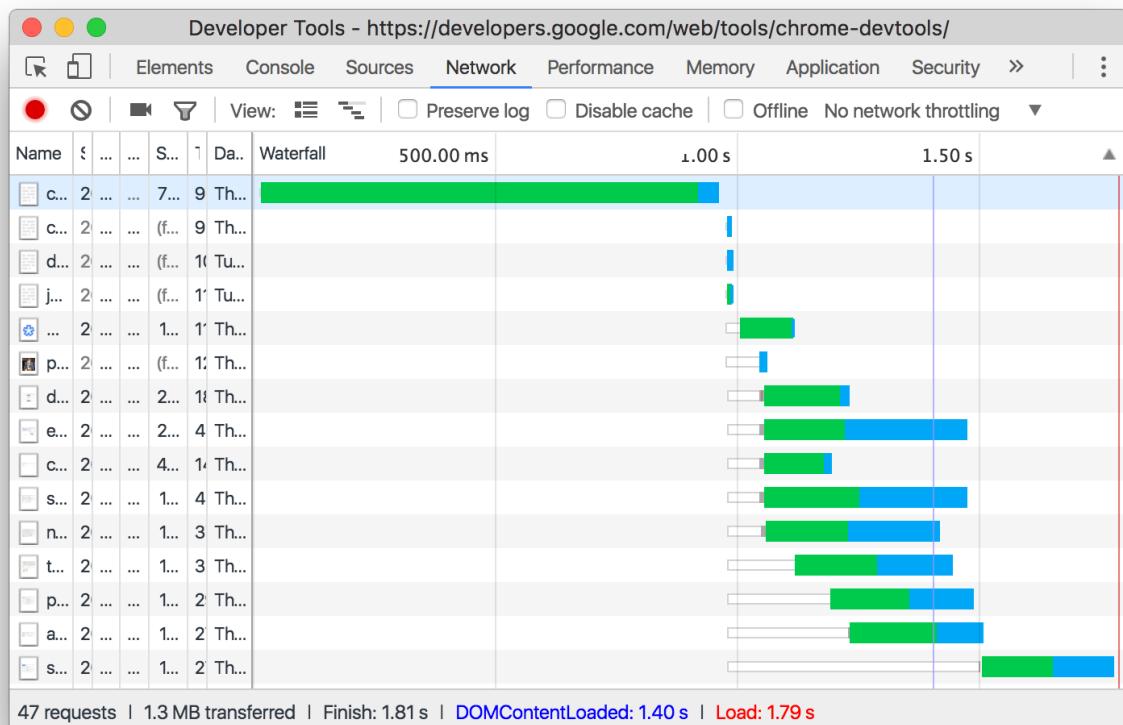


**Figure 18.** Adding a custom column to the Requests table.

## View the timing of requests in relation to one another

Use the Waterfall to view the timing of requests in relation to one another. By default, the Waterfall is organized by the start time of the requests. So, requests that are farther to the left started earlier than those that are farther to the right.

See [Sort by activity phase](#) to see the different ways that you can sort the Waterfall.



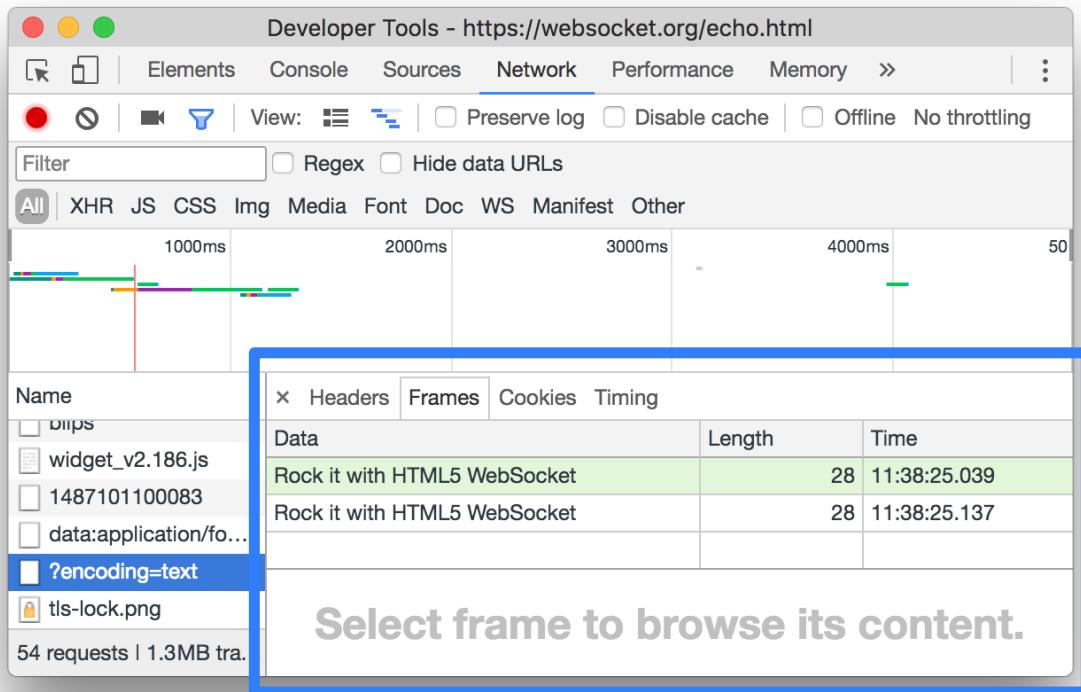
**Figure 19.** The Waterfall column of the Requests pane.

## Analyze the frames of a WebSocket Connection

To view the frames of a WebSocket connection:

1. Click the URL of the WebSocket connection, under the **Name** column of the Requests table.
2. Click the **Frames** tab. The table shows the last 100 frames.

To refresh the table, re-click the name of the WebSocket connection under the **Name** column of the Requests table.



**Figure 20.** The Frames tab, outlined in blue

The table contains three columns:

- **Data.** The message payload. If the message is plain text, it's displayed here. For binary opcodes, this column displays the opcode's name and code. The following opcodes are supported: Continuation Frame, Binary Frame, Connection Close Frame, Ping Frame, and Pong Frame.
- **Length.** The length of the message payload, in bytes.
- **Time.** The time when the message was received or sent.

Messages are color-coded according to their type:

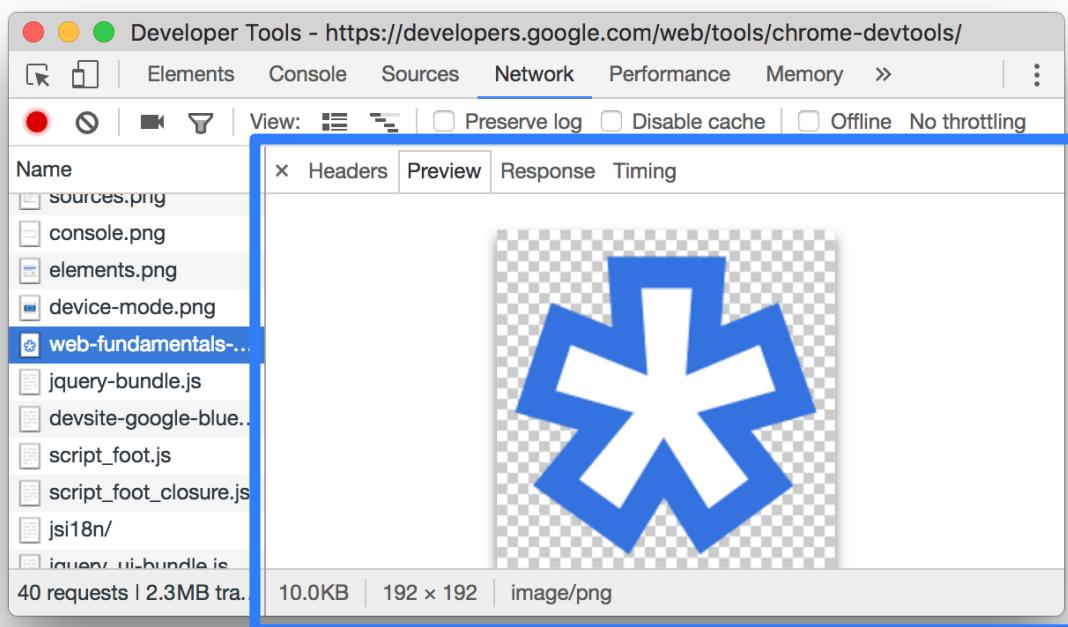
- Outgoing text messages are light-green.
- Incoming text messages are white.
- WebSocket opcodes are light-yellow.
- Errors are light-red.

## View a preview of a response body

To view a preview of a response body:

1. Click the URL of the request, under the **Name** column of the Requests table.
2. Click the **Preview** tab.

This tab is mostly useful for viewing images.

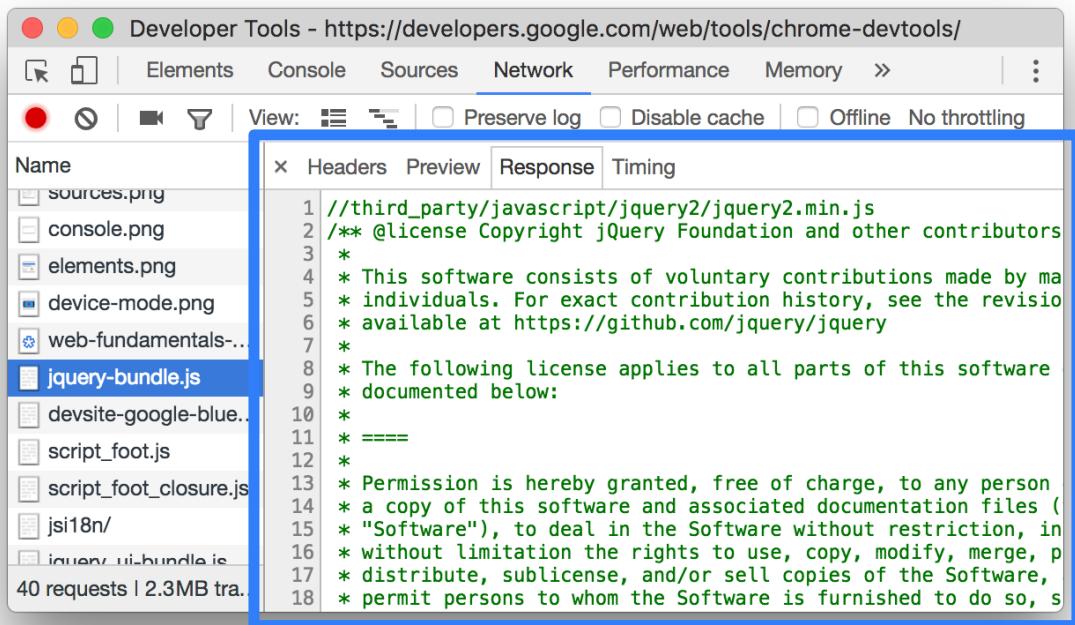


**Figure 21.** The Preview tab, outlined in blue

## View a response body

To view the response body to a request:

1. Click the URL of the request, under the **Name** column of the Requests table.
2. Click the **Headers** tab.

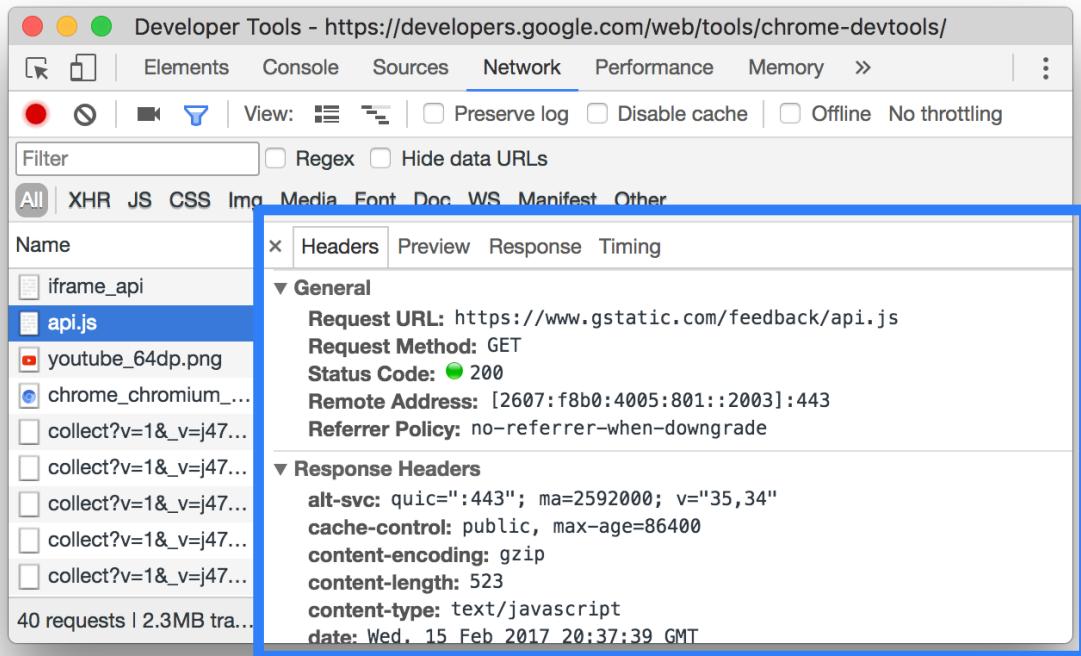


**Figure 22.** The Response tab, outlined in blue

## View HTTP headers

To view HTTP header data about a request:

1. Click on the URL of the request, under the **Name** column of the Requests table.
2. Click the **Headers** tab.



**Figure 23.** The Headers tab, outlined in blue

## View HTTP header source

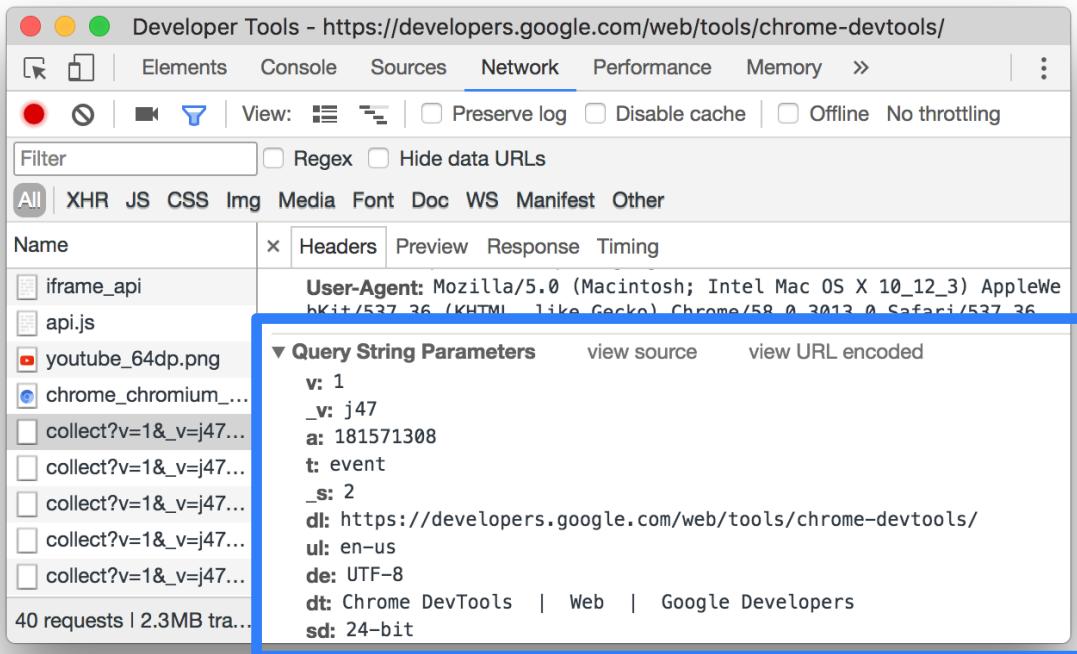
By default, the Headers tab shows header names alphabetically. To view the HTTP header names in the order they were received:

1. Open the **Headers** tab for the request you're interested in. See [View HTTP headers](#).
2. Click **view source**, next to the **Request Header** or **Response Header** section.

## View query string parameters

To view the query string parameters of a URL in a human-readable format:

1. Open the **Headers** tab for the request you're interested in. See [View HTTP headers](#).
2. Go to the **Query String Parameters** section.



**Figure 24.** The Query String Parameters section, outlined in blue

## View query string parameters source

To view the query string parameter source of a request:

1. Go to the Query String Parameters section. See [View query string parameters](#).
2. Click **view source**.

## View URL-encoded query string parameters

To view query string parameters in a human-readable format, but with encodings preserved:

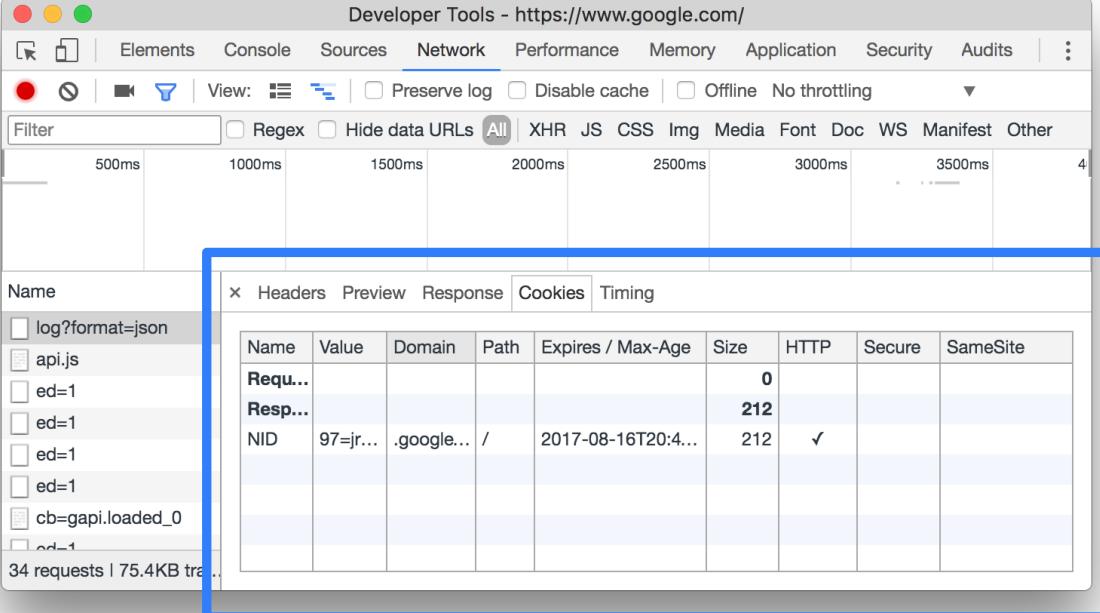
1. Go to the Query String Parameters section. See [View query string parameters](#).
2. Click **view URL encoded**.

## View cookies

To view the cookies sent in a request's HTTP header:

1. Click the URL of the request, under the **Name** column of the Requests table.
2. Click the **Cookies** tab.

See [Fields](#) for a description of each of the columns.



The screenshot shows the Network tab of the Developer Tools for the URL https://www.google.com/. A blue box highlights the 'Cookies' tab within the sub-table. The sub-table has columns: Name, Value, Domain, Path, Expires / Max-Age, Size, HTTP, Secure, and SameSite. One row is visible for the cookie 'NID' with the value '97=jr...', domain '.google...', path '/', expiration '2017-08-16T20:4...', size '212', and checked 'HTTP' and 'Secure'.

Name	Value	Domain	Path	Expires / Max-Age	Size	HTTP	Secure	SameSite
NID	97=jr...	.google...	/	2017-08-16T20:4...	212	✓		

**Figure 25.** The Cookies tab, outlined in blue

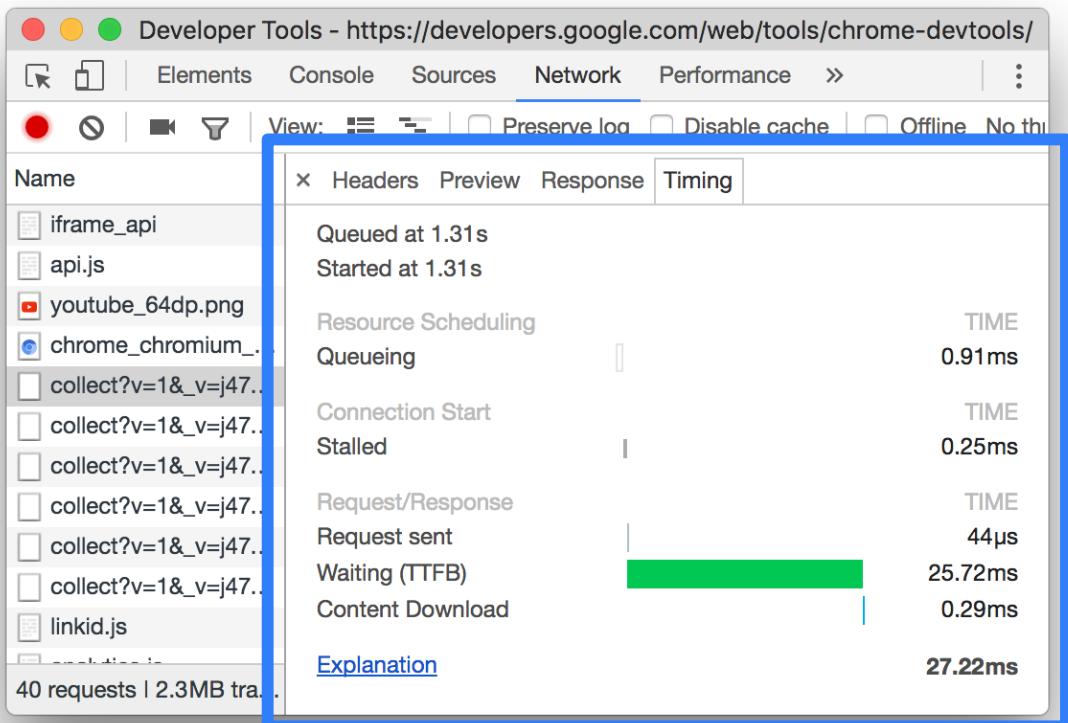
## View the timing breakdown of a request

To view the timing breakdown of a request:

1. Click the URL of the request, under the **Name** column of the Requests table.
2. Click the **Timing** tab.

See [Preview a timing breakdown](#) for a faster way to access this data.

See [Timing breakdown phases explained](#) for more information about each of the phases that you may see in the Timing tab.



**Figure 26.** The Timing tab, outlined in blue

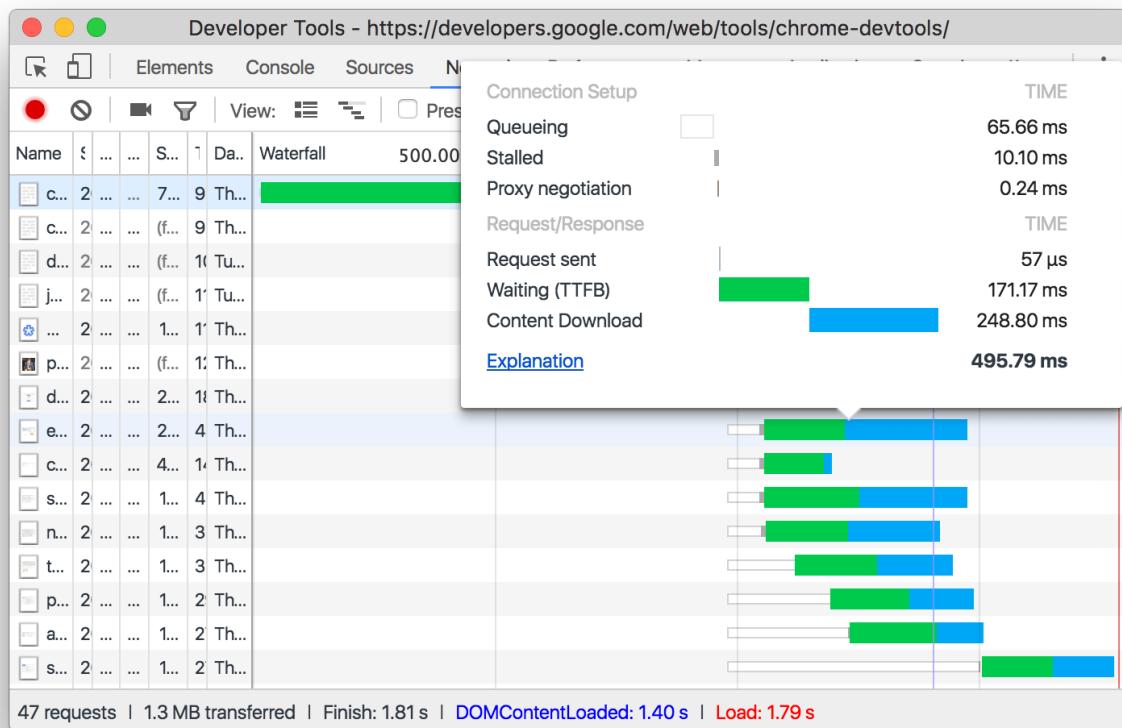
Here's more information about each of the phases.

See [View timing breakdown](#) for another way to access this view.

### Preview a timing breakdown

To view a preview of the timing breakdown of a request, hover over the request's entry in the **Waterfall** column of the Requests table.

See [View the timing breakdown of a request](#) for a way to access this data that does not require hovering.



**Figure 27.** Previewing the timing breakdown of a request

### Timing breakdown phases explained

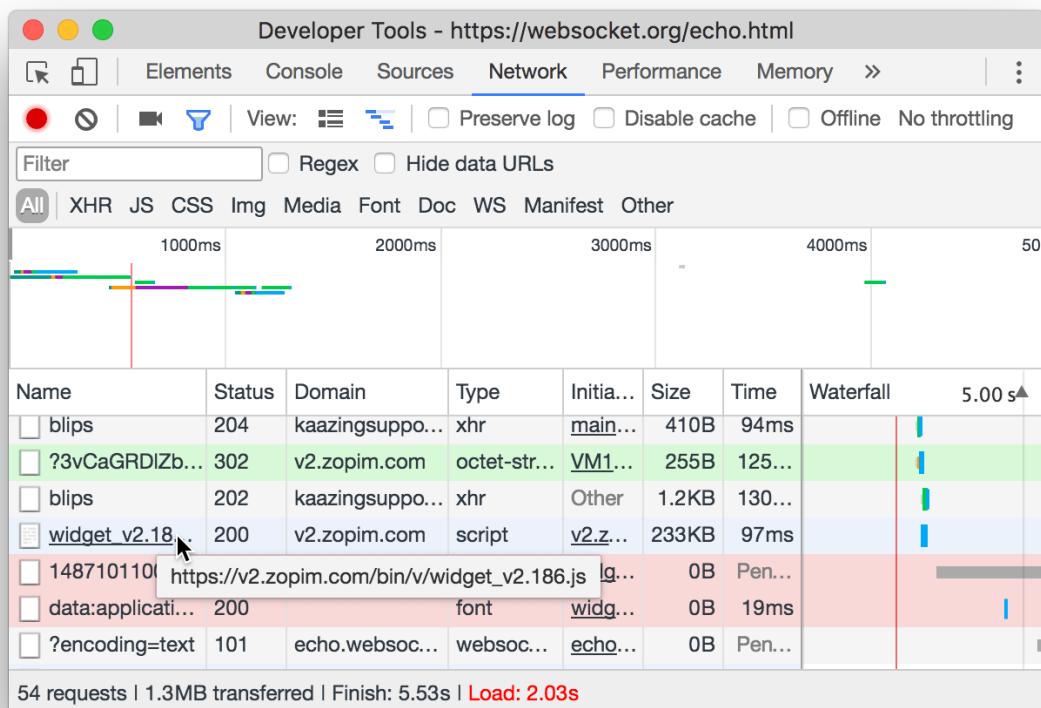
Here's more information about each of the phases you may see in the Timing tab:

- **Queueing.** The browser queues requests when:
  - There are higher priority requests.
  - There are already six TCP connections open for this origin, which is the limit. Applies to HTTP/1.0 and HTTP/1.1 only.
  - The browser is briefly allocating space in the disk cache
- **Stalled.** The request could be stalled for any of the reasons described in **Queueing**.
- **DNS Lookup.** The browser is resolving the request's IP address.
- **Proxy negotiation.** The browser is negotiating the request with a proxy server.
- **Request sent.** The request is being sent.
- **ServiceWorker Preparation.** The browser is starting up the service worker.
- **Request to ServiceWorker.** The request is being sent to the service worker.

- **Waiting (TTFB)**. The browser is waiting for the first byte of a response. TTFB stands for Time To First Byte. This timing includes 1 round trip of latency and the time the server took to prepare the response.
- **Content Download**. The browser is receiving the response.
- **Receiving Push**. The browser is receiving data for this response via HTTP/2 Server Push.
- **Reading Push**. The browser is reading the local data previously received.

## View initiators and dependencies

To view the initiators and dependencies of a request, hold Shift and hover over the request in the Requests table. DevTools colors initiators green, and dependencies red.

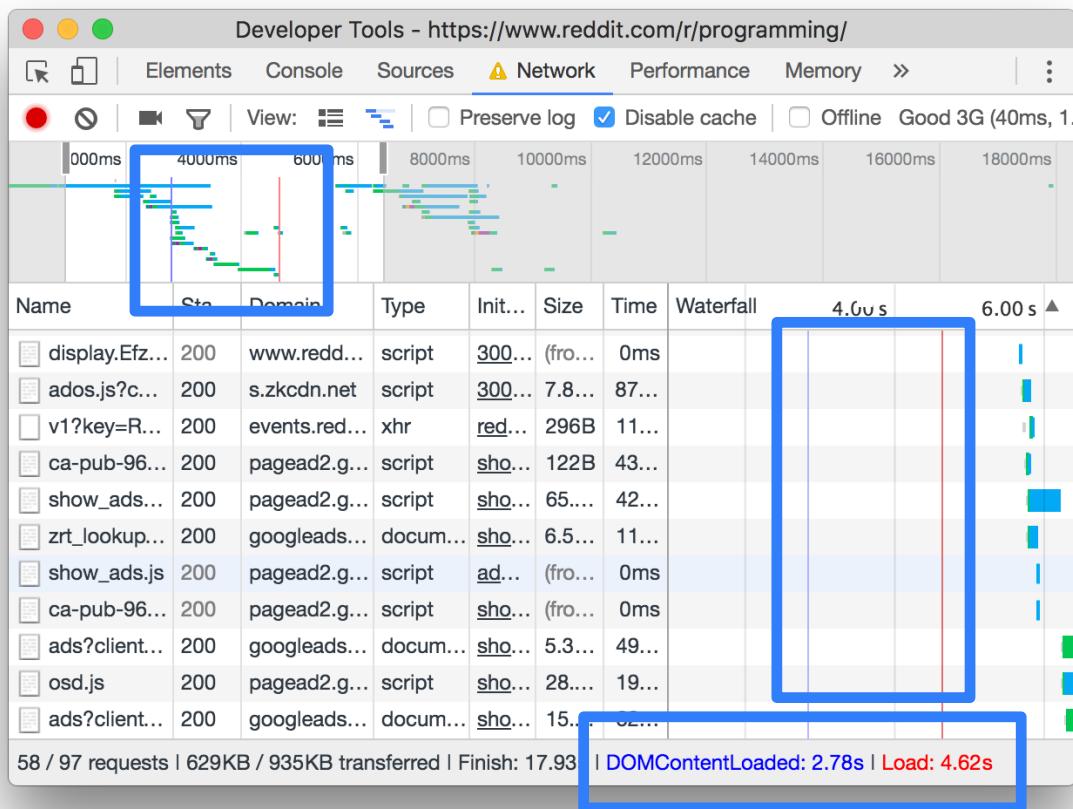


**Figure 28.** Viewing the initiators and dependencies of a request

When the Requests table is ordered chronologically, the first green request above the request that you're hovering over is the initiator of the dependency. If there's another green request above that, that higher request is the initiator of the initiator. And so on.

## View load events

DevTools displays the timing of the `DOMContentLoaded` and `load` events in multiple places on the Network panel. The `DOMContentLoaded` event is colored blue, and the `load` event is red.

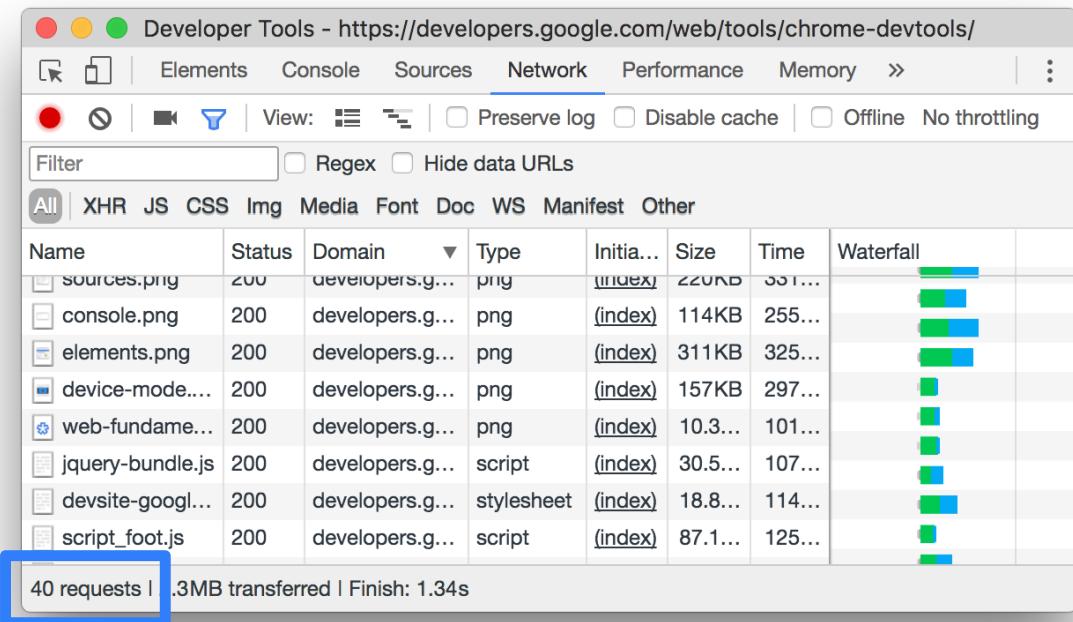


**Figure 29.** The locations of the `DOMContentLoaded` and `load` events in the Network panel

## View the total number of requests

The total number of requests is listed in the Summary pane, at the bottom of the Network panel.

**Caution:** This number only tracks requests that have been logged since DevTools was opened. If other requests occurred before DevTools was opened, those requests aren't counted.

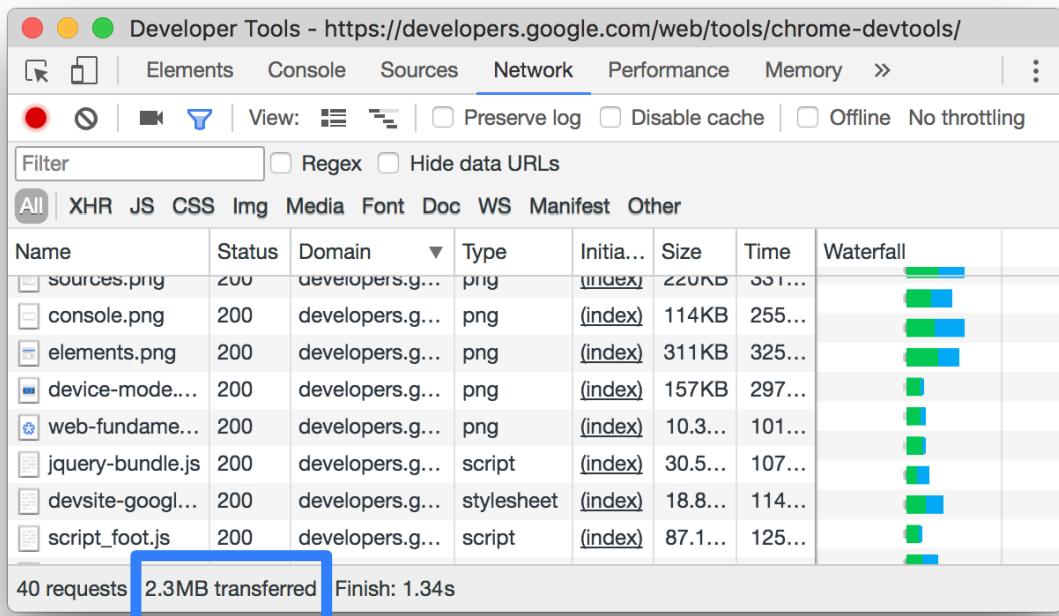


**Figure 30.** The total number of requests since DevTools was opened

## View the total download size

The total download size of requests is listed in the Summary pane, at the bottom of the Network panel.

**Caution:** This number only tracks requests that have been logged since DevTools was opened. If other requests occurred before DevTools was opened, those requests aren't counted.

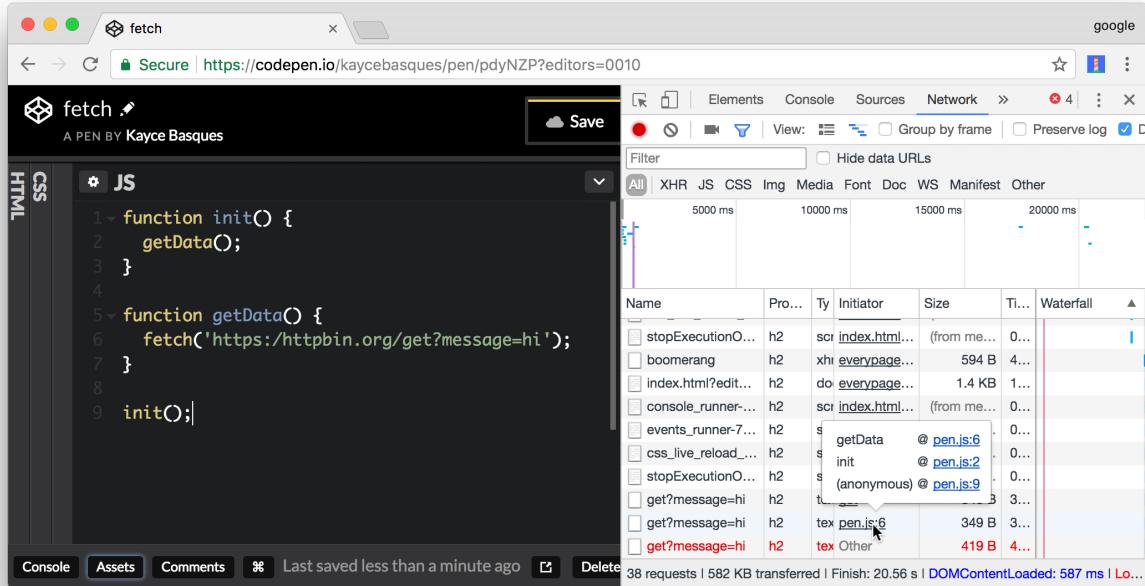


**Figure 31.** The total download size of requests

See [View the uncompressed size of a resource](#) to see how large resources are after the browser uncompresses them.

## View the stack trace that caused a request

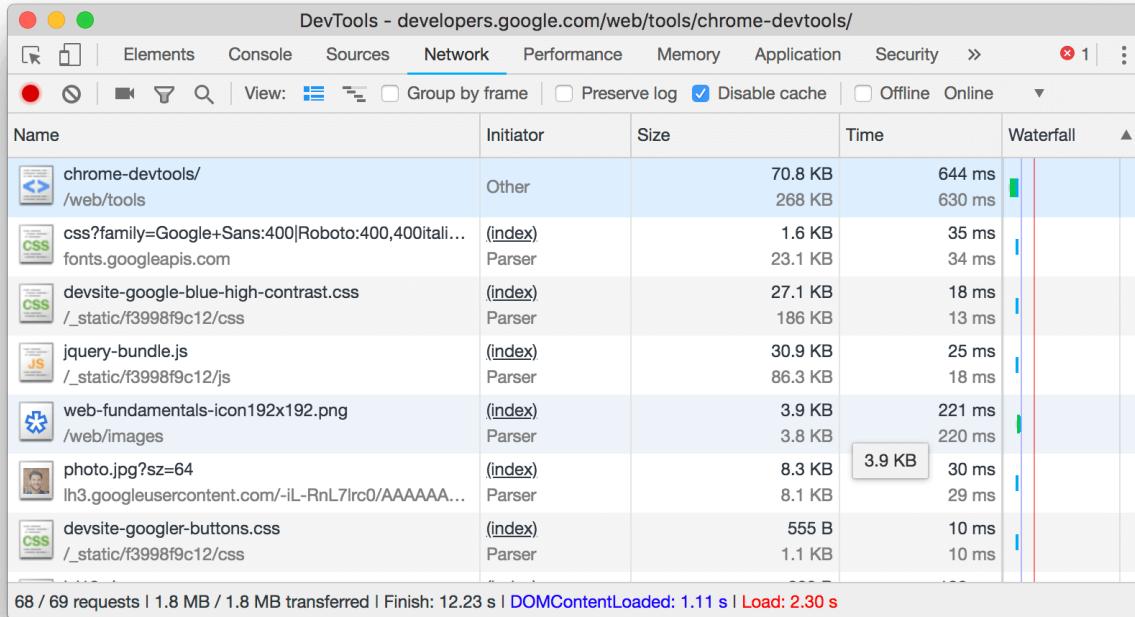
When a JavaScript statement causes a resource to be requested, hover over the **Initiator** column to view the stack trace leading up to the request.



**Figure 32.** The stack trace leading up to a resource request

## View the uncompressed size of a resource

Click **Use Large Request Rows** and then look at the bottom value of the **Size** column.



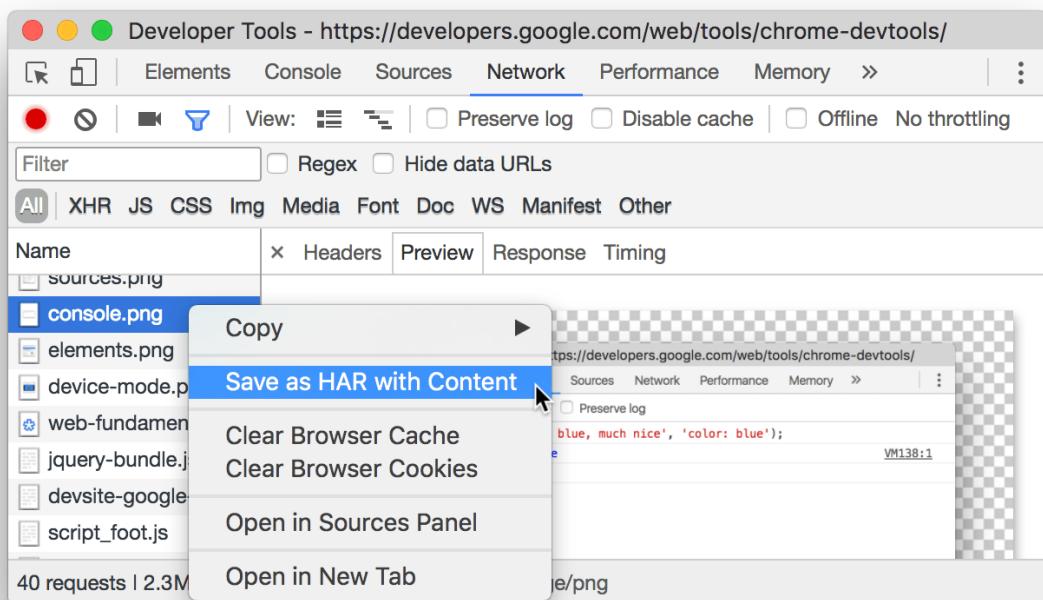
**Figure 33.** The compressed size of the `jquery-bundle.js` file that was sent over the network was **30.9 KB**, whereas the uncompressed size was **86.3 KB**

## Export requests data

### Save a request as HAR with content

To save a request in the HAR format with content:

1. Right-click the row containing the request in the Requests table.
2. Select **Save as HAR with Content**.

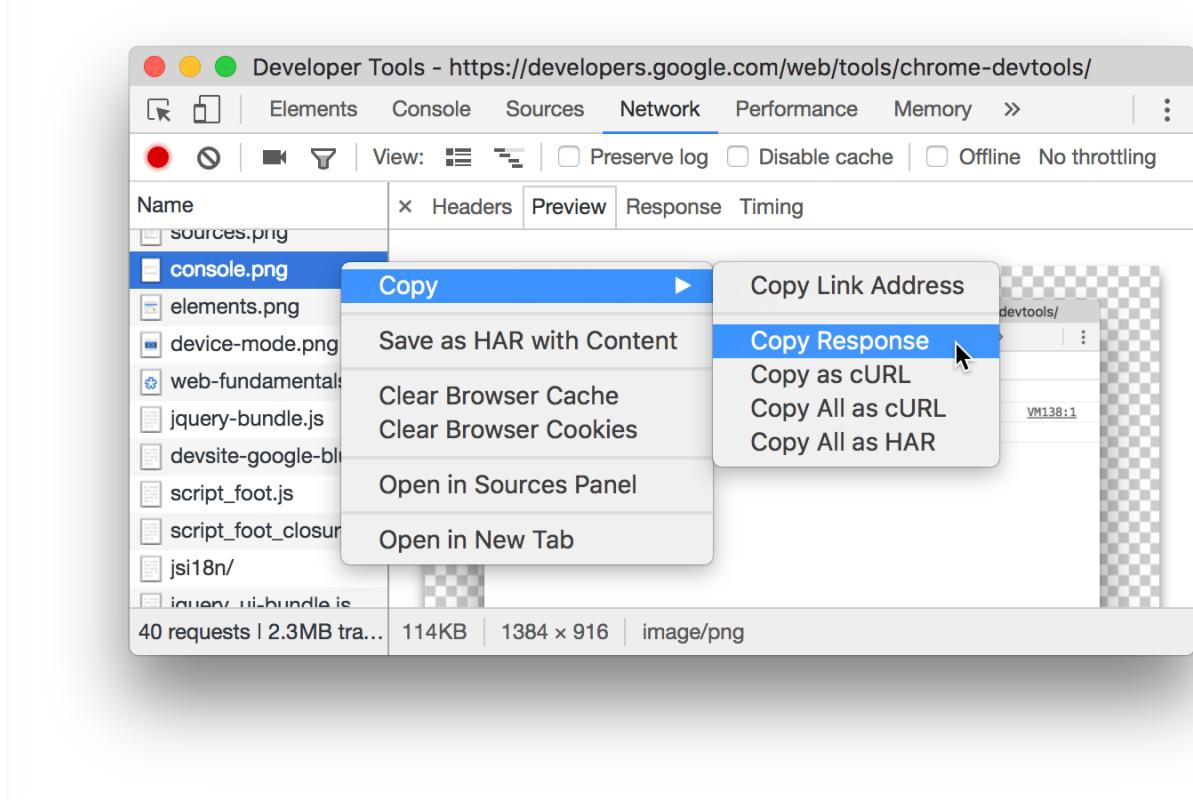


**Figure 34.** Selecting Save As HAR With Content

### Copy one or more requests to the clipboard

Under the **Name** column of the Requests table, right-click a request, hover over **Copy**, and select one of the following options:

- **Copy Link Address.** Copy the request's URL to the clipboard.
- **Copy Response.** Copy the response body to the clipboard.
- **Copy as cURL.** Copy the request as a cURL command.
- **Copy All as cURL.** Copy all requests as a chain of cURL commands.
- **Copy All as HAR.** Copy all requests as HAR data.



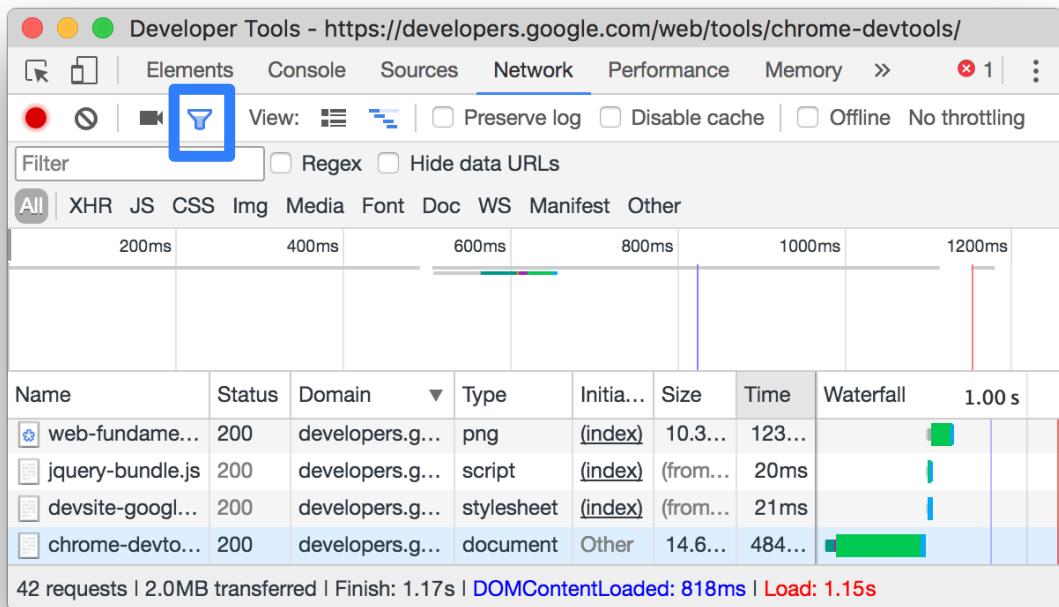
**Figure 35.** Selecting Copy Response

## Change the layout of the Network panel

Expand or collapse sections of the Network panel UI to focus on what's important to you.

## Hide the Filters pane

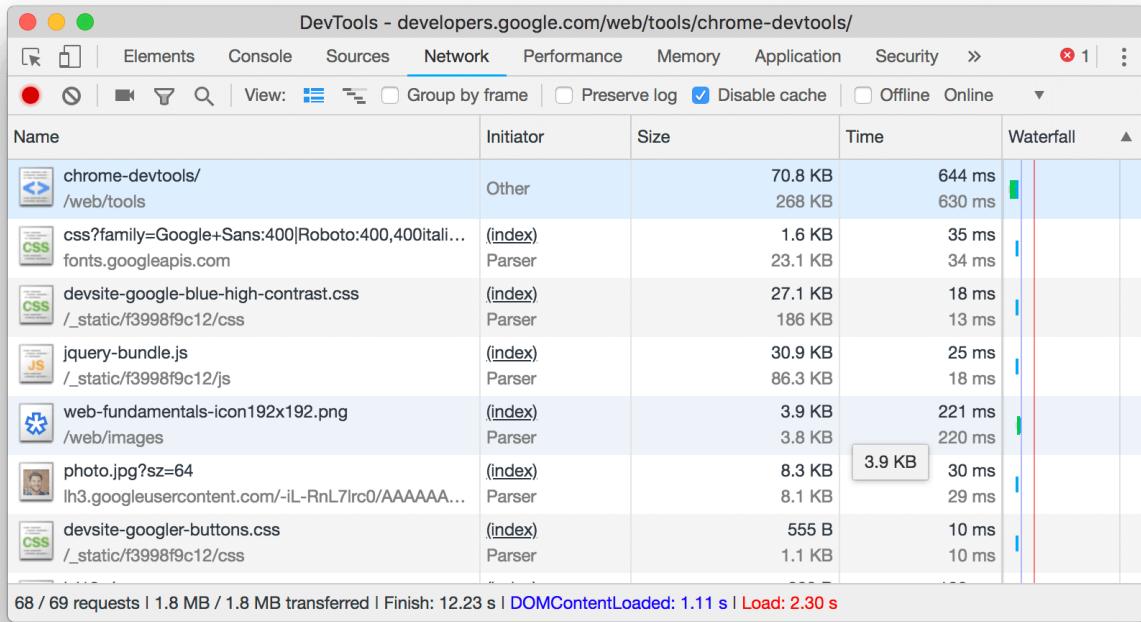
By default, DevTools shows the Filters pane. Click **Filter** to hide it.



**Figure 36.** Hide Filters, outlined in blue

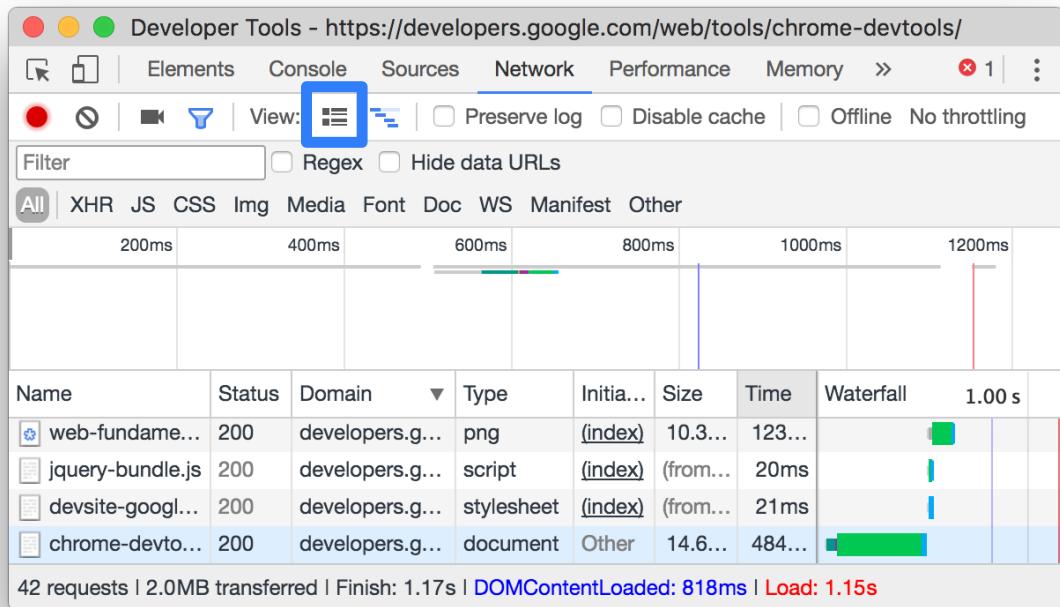
## Use large request rows

Use large rows when you want more whitespace in your network requests table. Some columns also provide a little more information when using large rows. For example, the bottom value of the **Size** column is the uncompressed size of a request.



**Figure 37.** An example of large request rows in the Requests pane

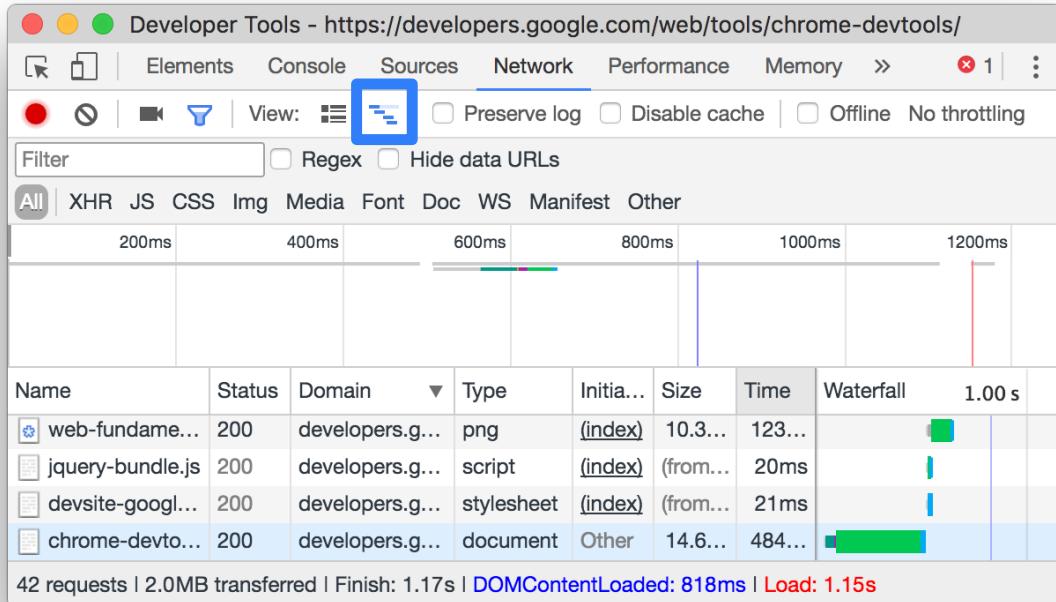
Click **Use large request rows** to enable large rows.



**Figure 38.** Large Request Rows, outlined in blue

## Hide the Overview pane

By default, DevTools shows the [Overview pane](#). Click **Hide overview**  to hide it.



**Figure 39.** Hide Overview, outlined in blue

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.