Google Chrome Developer Tools – Part 2

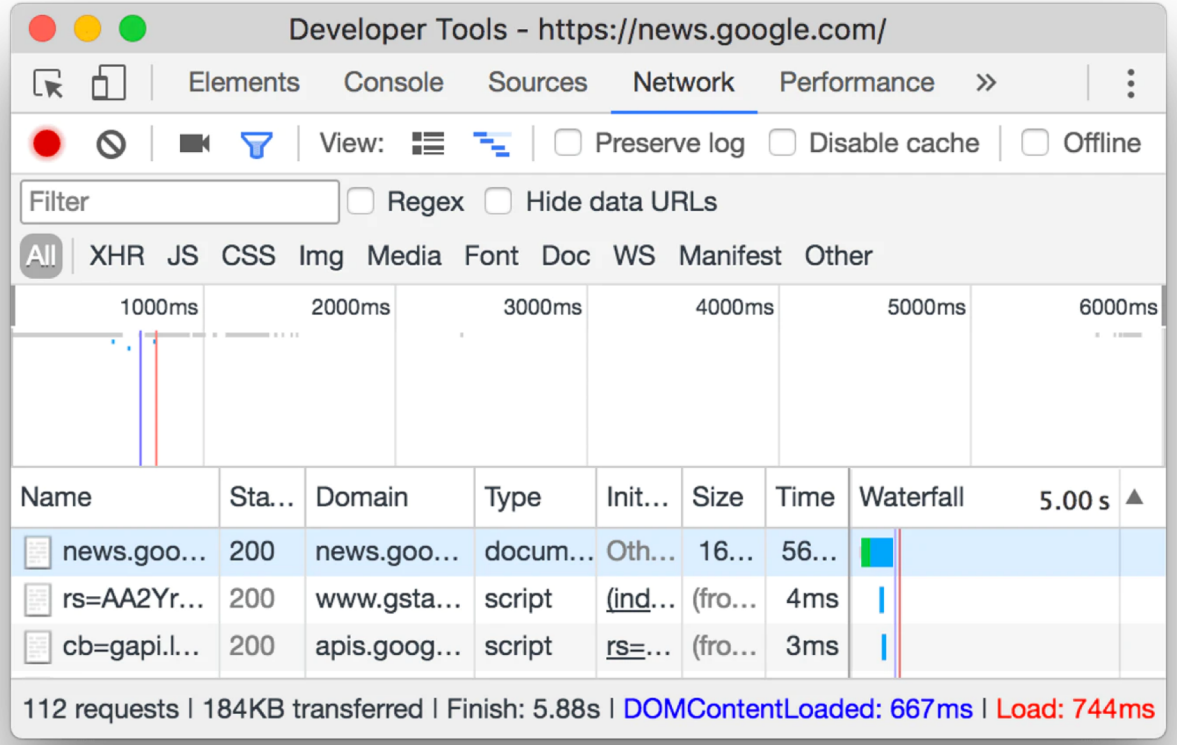
Network, Timeline, Profiles, Resources, Security, Audits

Networks Panel

Inspecting a page’s network activity

Networks Panel

- Helps you to analyze your page's loading behavior
- By default, DevTools records all network requests in the **Network** panel, so long as DevTools is open.



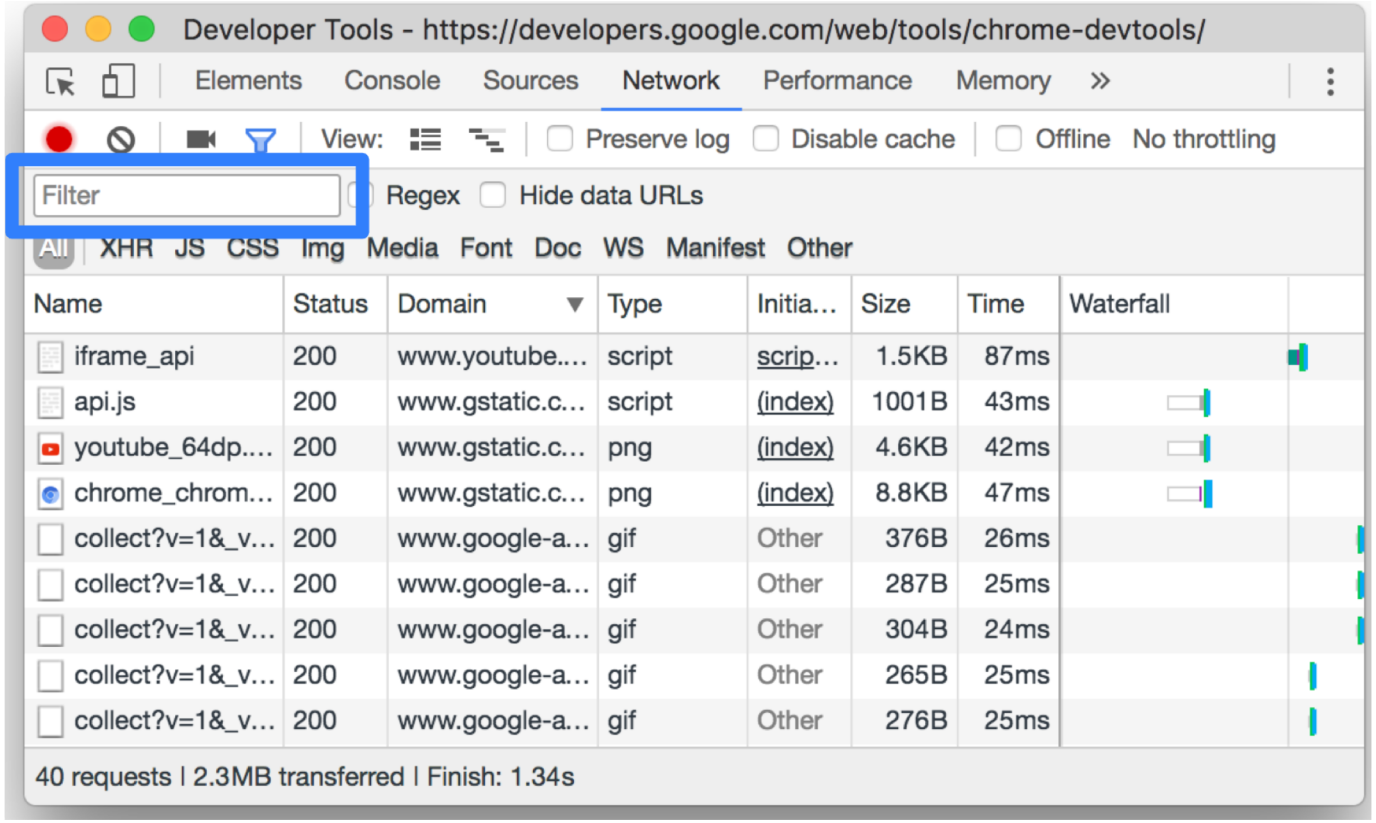
When to use

- In general, use the Network panel **when you need to make sure that resources are being downloaded or uploaded as expected.**

- The most common use cases for the Network panel are:
  - Making sure that resources are actually being uploaded or downloaded at all.
  - Inspecting the properties of an individual resource, such as its HTTP headers, content, size, and more.

Filter requests

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



- 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.

Important supported properties

- **cookie-name**
- **cookie-path**
- **cookie-value**
- **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.
- **has-response-header**. Show the resources that contain the specified HTTP response header.
- **url**. Show the resources that have a url matching the specified value.

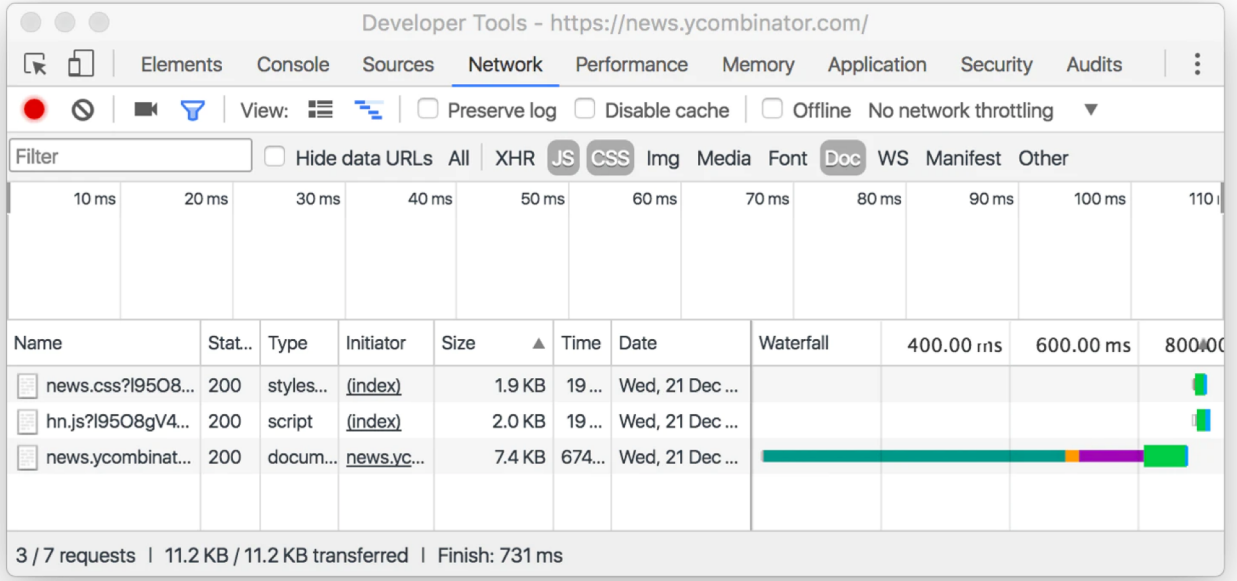
Important supported properties

- **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.
- **mime-type**. Show resources of a specified MIME type.
- **resource-type**. Show resources of a resource type, e.g. image.
- **scheme**. Show resources retrieved over unprotected HTTP (**scheme:http**) or protected HTTPS (**scheme:https**).
- **status-code**. Only show resources whose HTTP status code match the specified code.

Filter requests by type

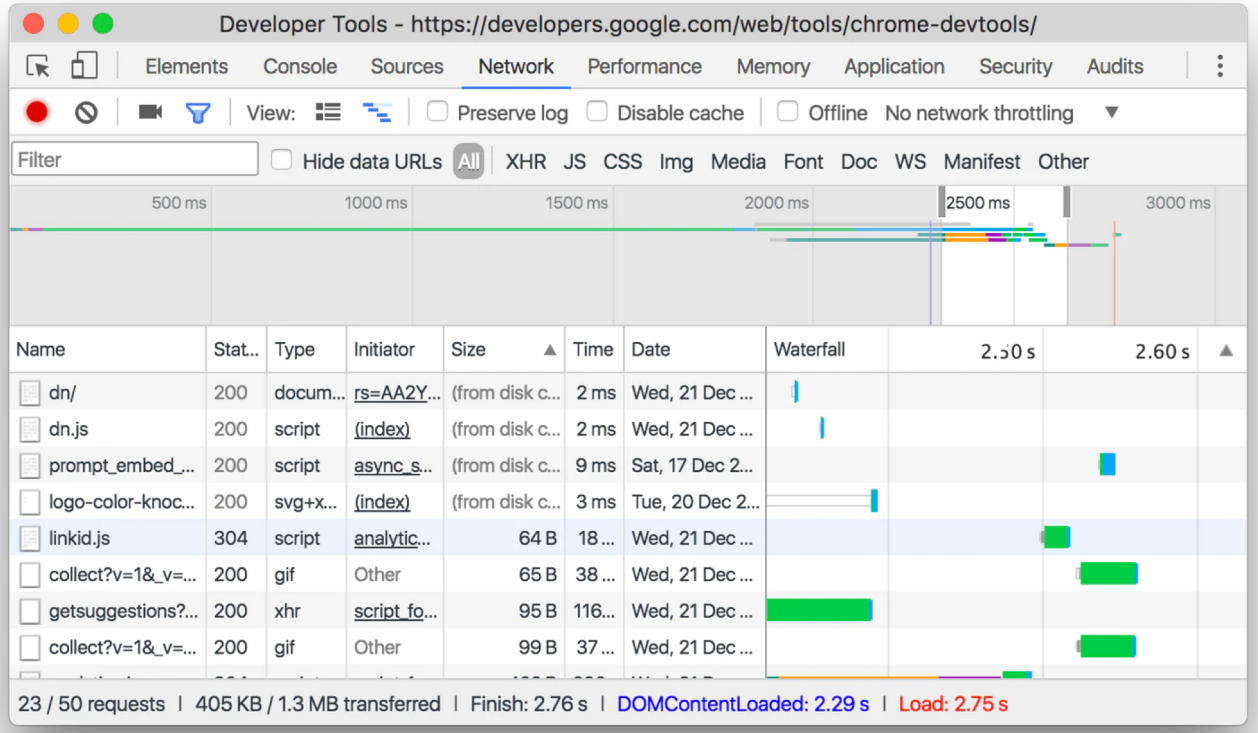
- To filter requests by request type, click the **XHR, JS, CSS, Img, Media, Font, Doc, WS (WebSocket), Manifest, or Other (any other type not listed here)** buttons on the Network panel.

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



Filter requests by time

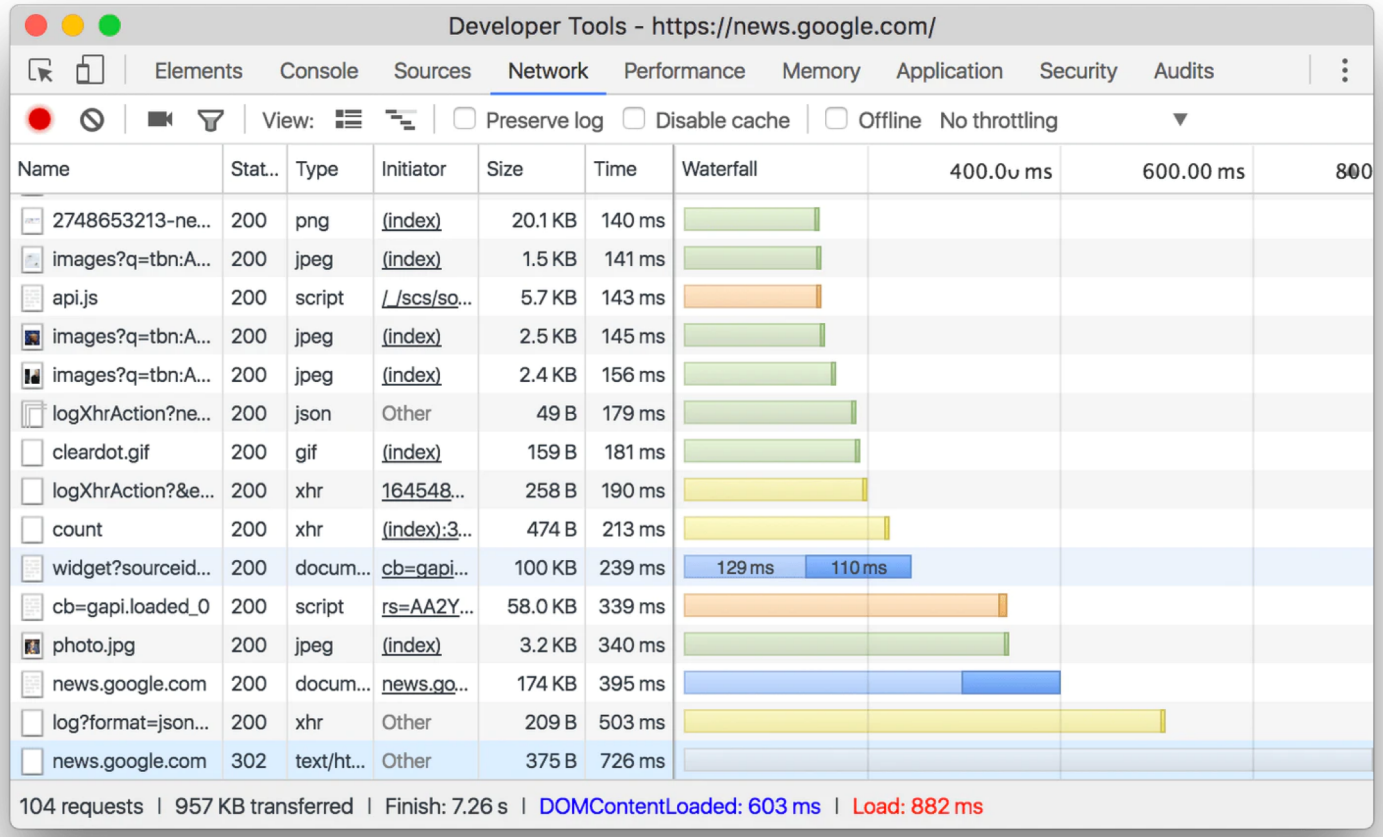
- Click and drag left or right on the **Overview** pane to only display requests that were active during that time frame.
- The filter is inclusive.
- Any request that was active during the highlighted time is shown.



Sort requests

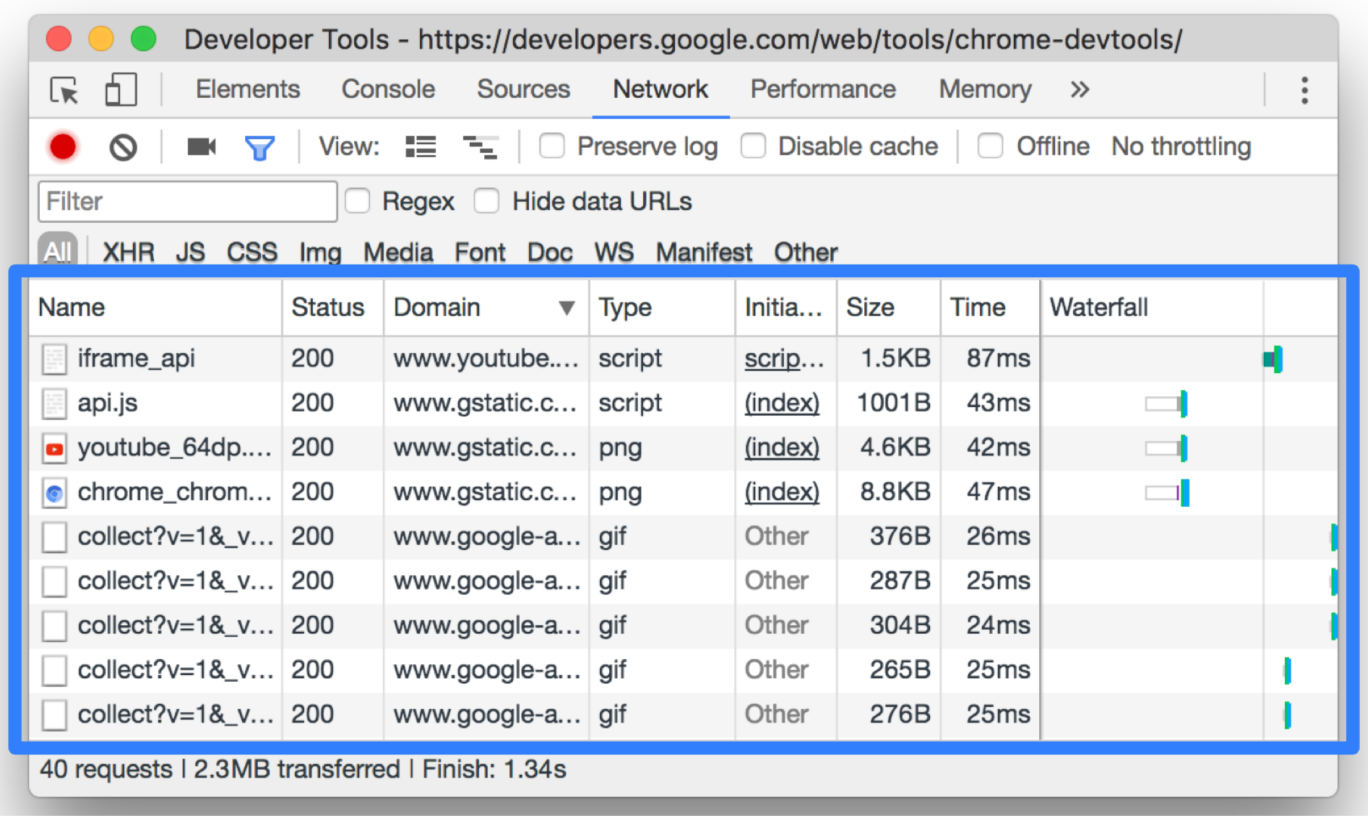
By default, the requests in the Requests table are sorted by initiation time, but you can sort the table using other criteria.

- **By column**
  - **Status**
  - **Type**
  - **Initiator**
  - **Size**
  - **Time**
- **By activity phase (Waterfall)**
  - **Start Time**
  - **Response Time**
  - **End Time**
  - **Total Duration**
  - **Latency**



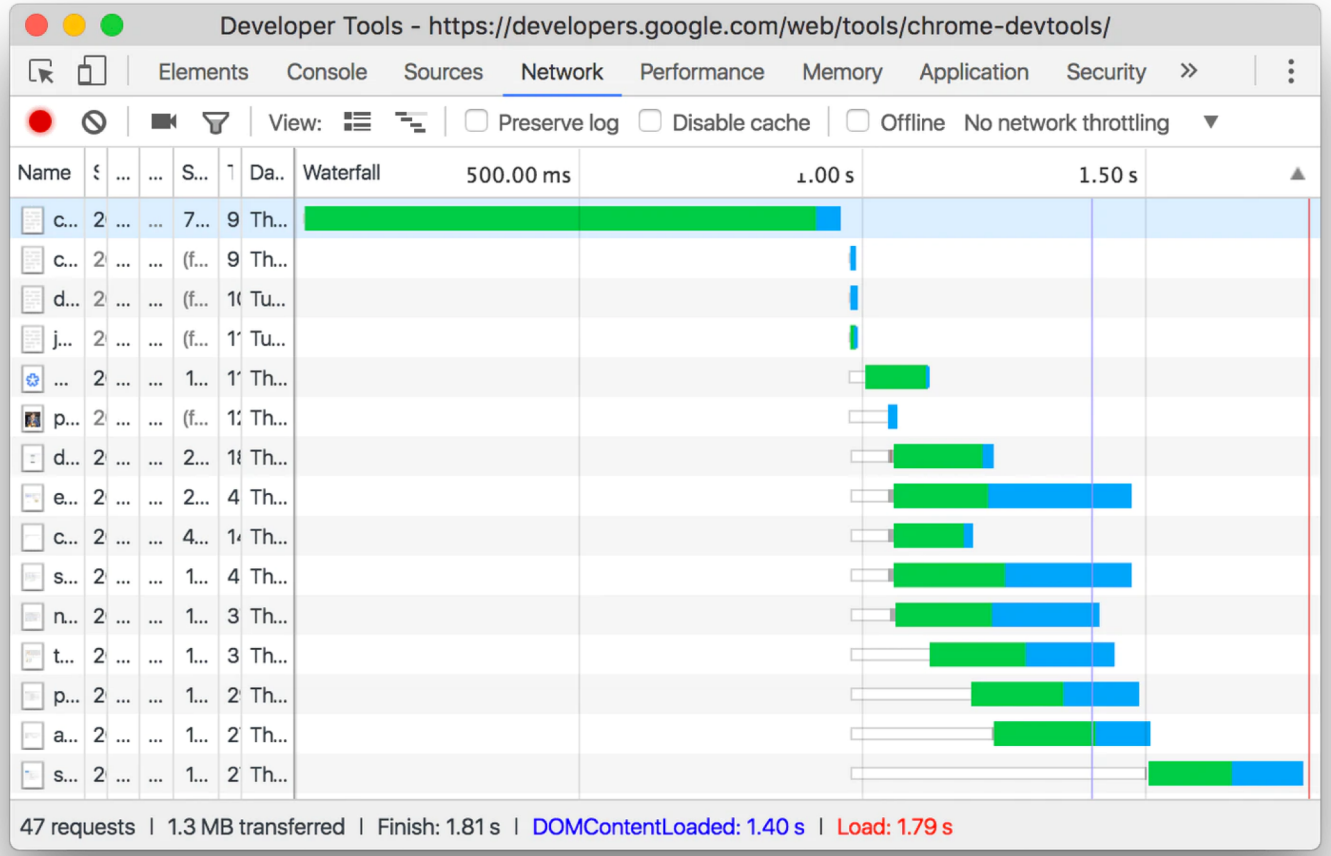
Analyze requests

View a log of requests: Clicking or hovering over requests reveals more information about them.



View the timing of requests in relation to one another

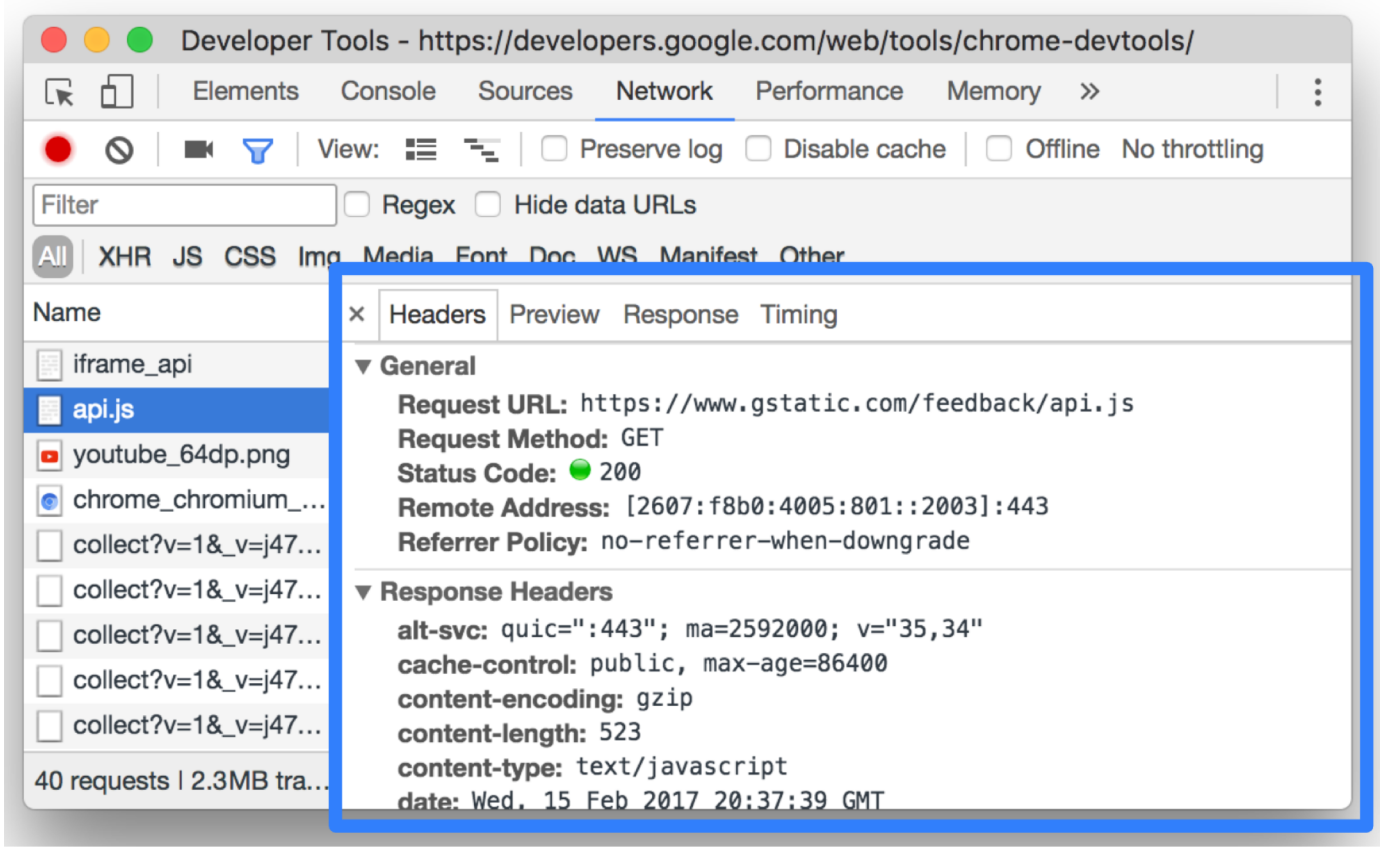
- Use the Waterfall to view the timing of requests in relation to one another.
- By default, it is organized by the **start time** of the requests
- That means that requests that are farther to the left **started earlier** than those that are farther to the right.



View Requests detailed information

Click the URL of the request, under the **Name** column of the Requests (or Response) table to see:

- A preview of response body
- Full response body
- HTTP Headers data
- Cookies
- The timing breakdown



Stop recording network requests

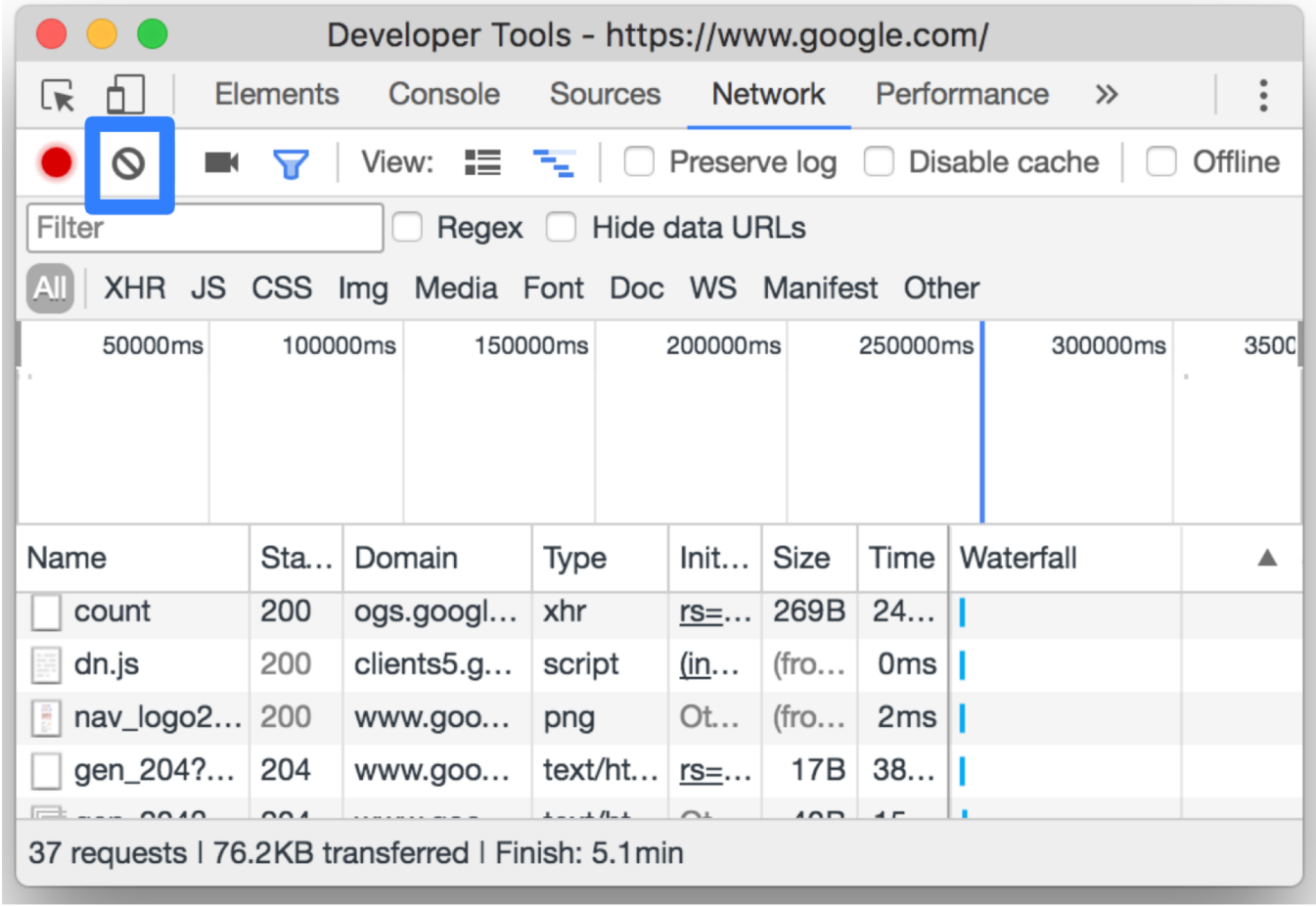
- Click **Stop recording network log** button on the Network panel

- Or press **Command+E (Mac)** or **Control+E (Windows, Linux)** while the Network panel is in focus

In any case, it will change the icon from **red** to **gray**.

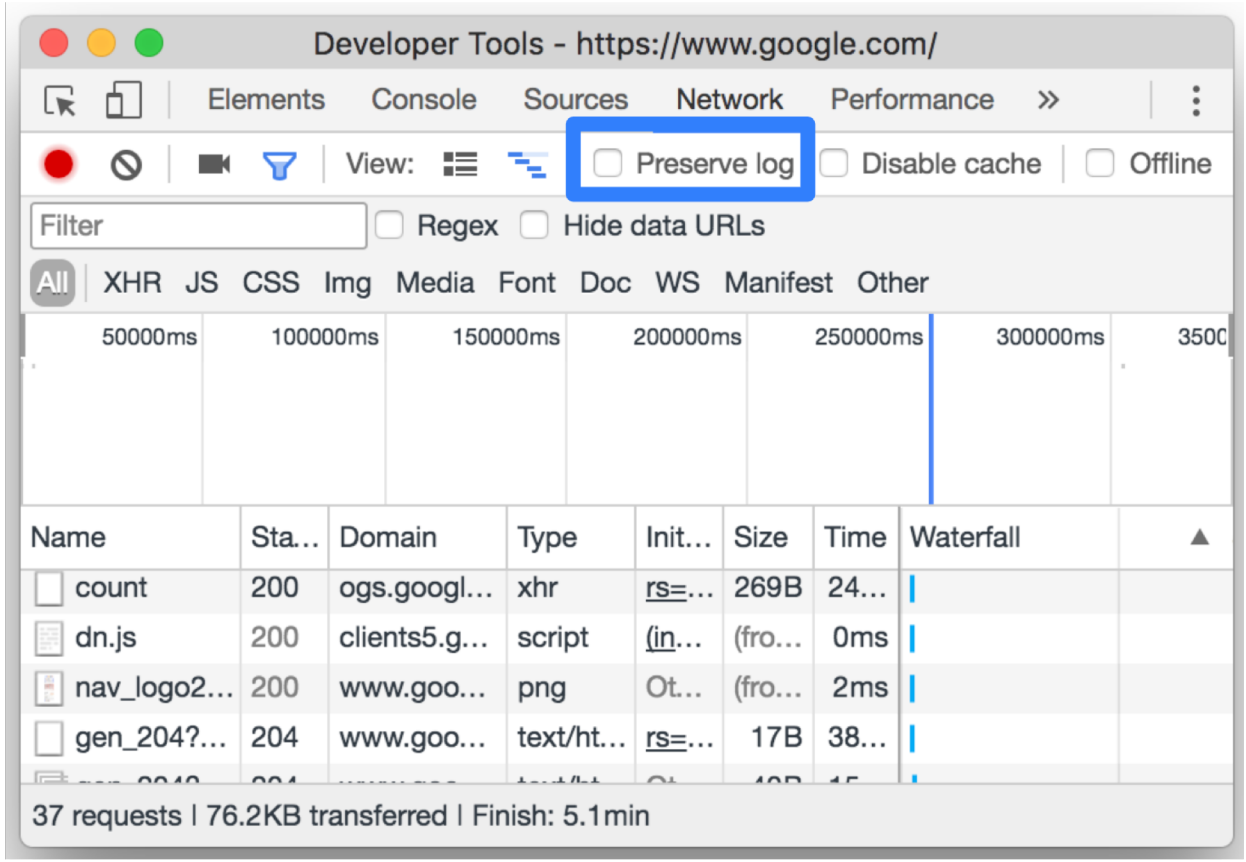
Clear requests

Click **Clear** on the Network panel to clear all requests from the Requests table.



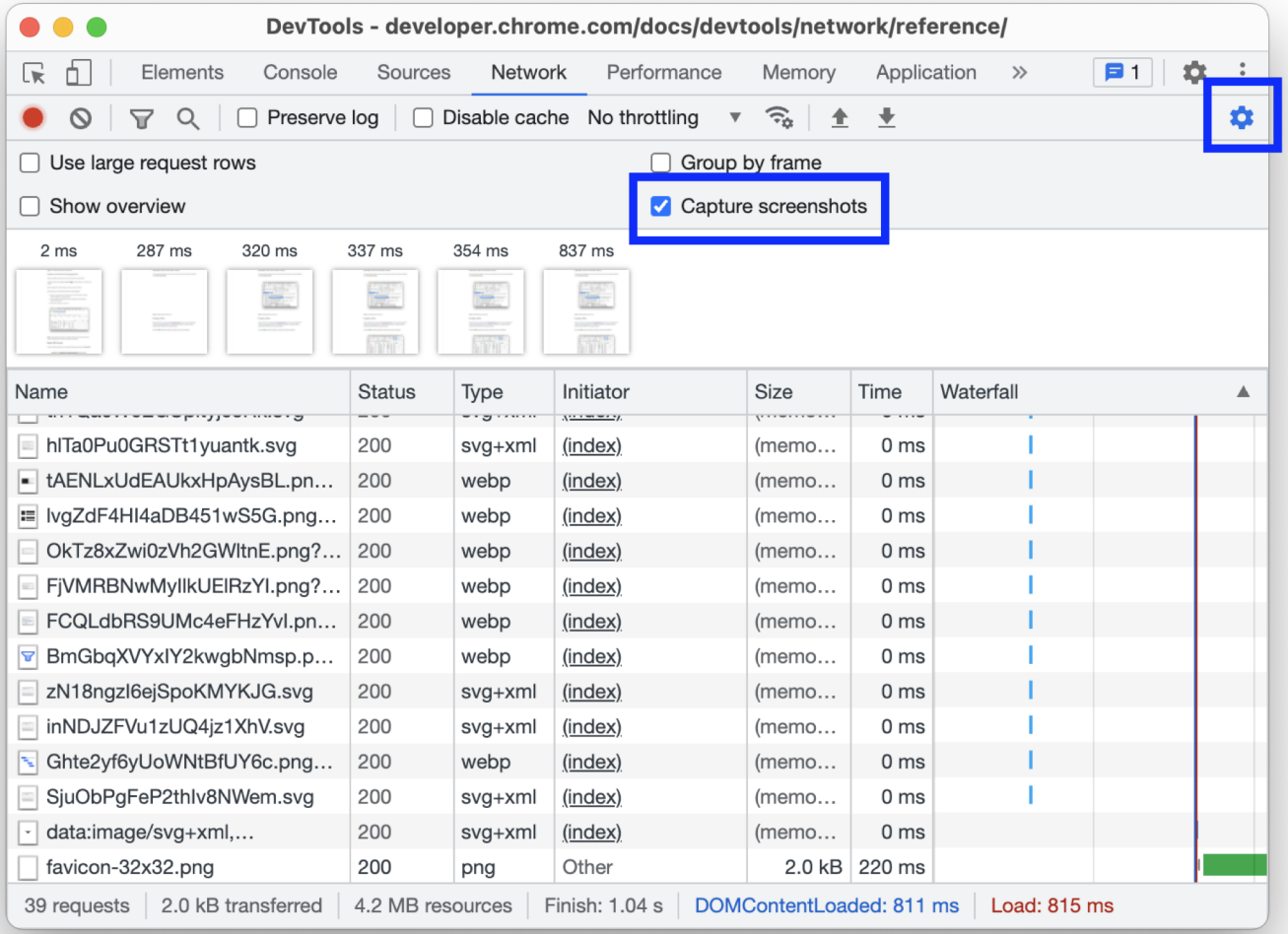
Save requests across page loads

- Check the **Preserve log** checkbox on the Network panel.
- DevTools saves all requests until you disable **Preserve log**.



Capture screenshots during page load

- Capture screenshots to analyze what users see as they wait for your page to load.
- To enable screenshots, open **Settings** inside the Network panel and check **Capture screenshots**.



Capture screenshots during page load

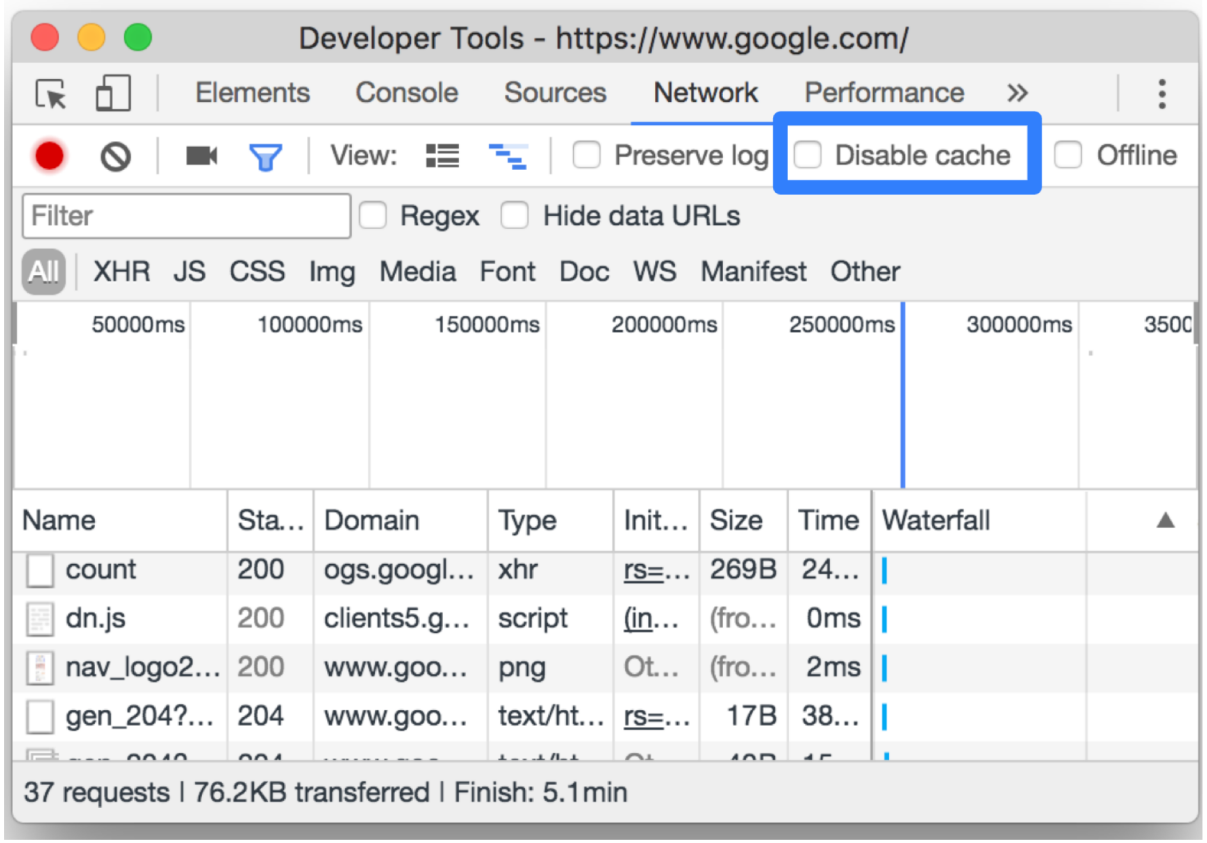
- **Reload the page** while the Network panel is in focus to capture screenshots.

- Once captured, you can interact with screenshots:
  - **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.

Emulate a first-time visitor

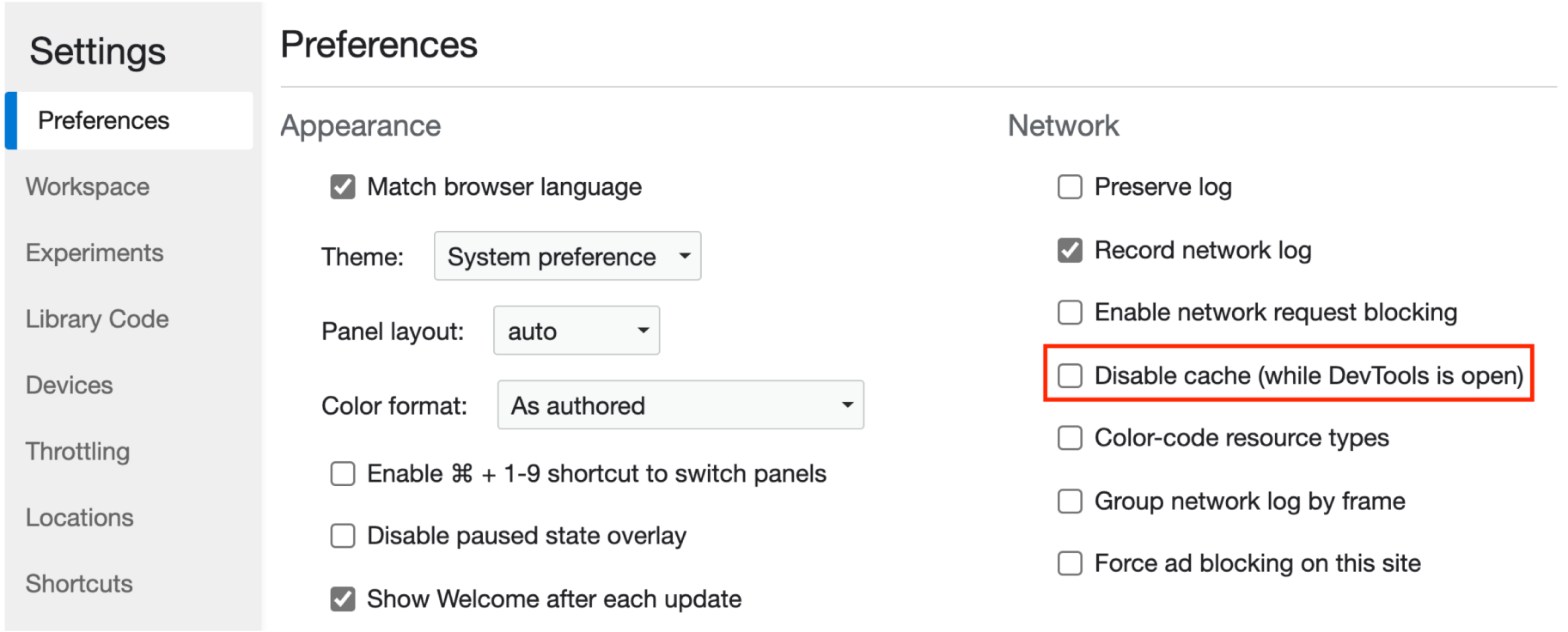
To emulate how a first-time user experiences your site, check the **Disable cache** checkbox.

Remember that requests are served from the browser cache on repeat visits.

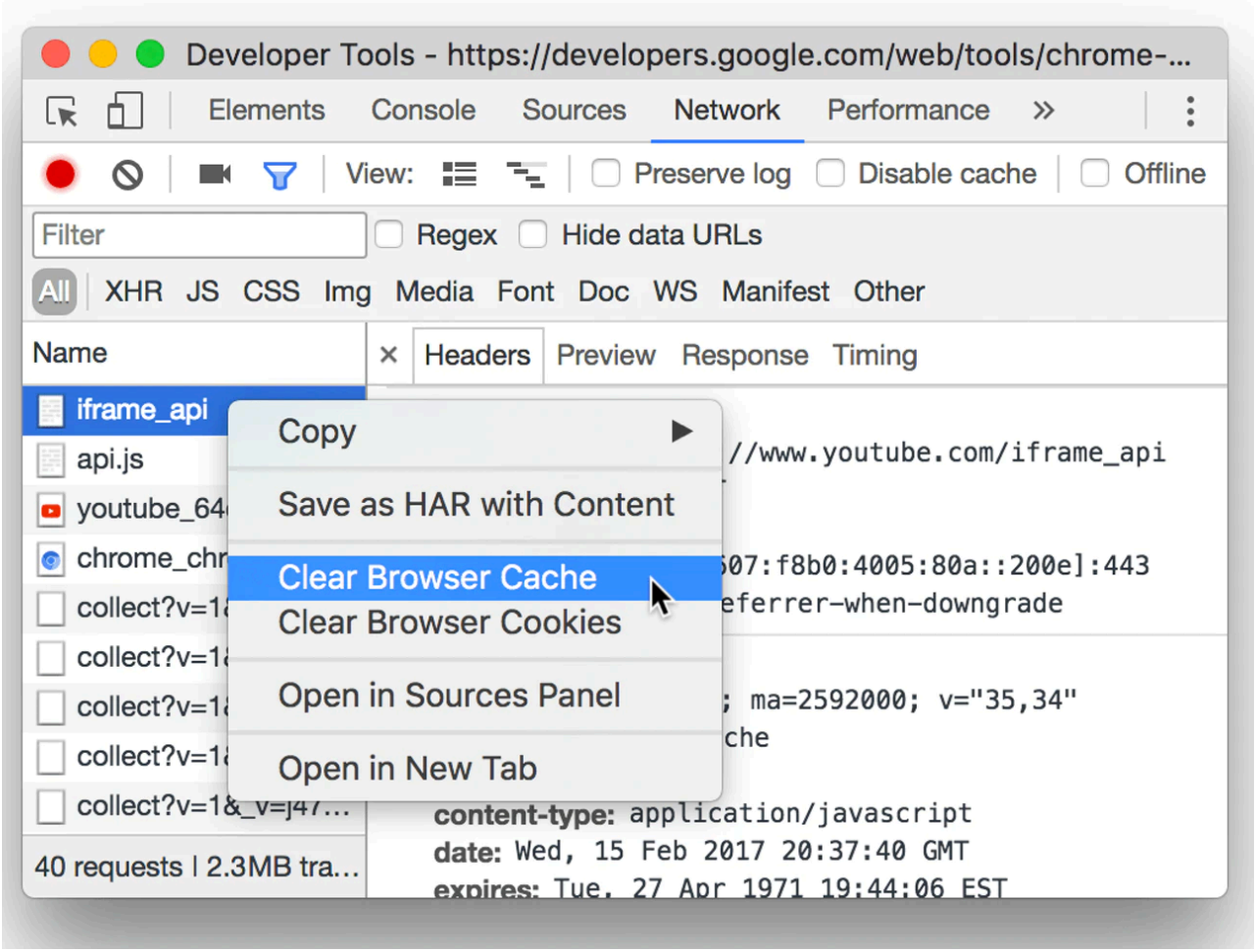


If you want to **disable the cache while working in other DevTools panels**, set the Network preferences.

- Open Settings → Preferences.
- Check the **Disable cache** checkbox.

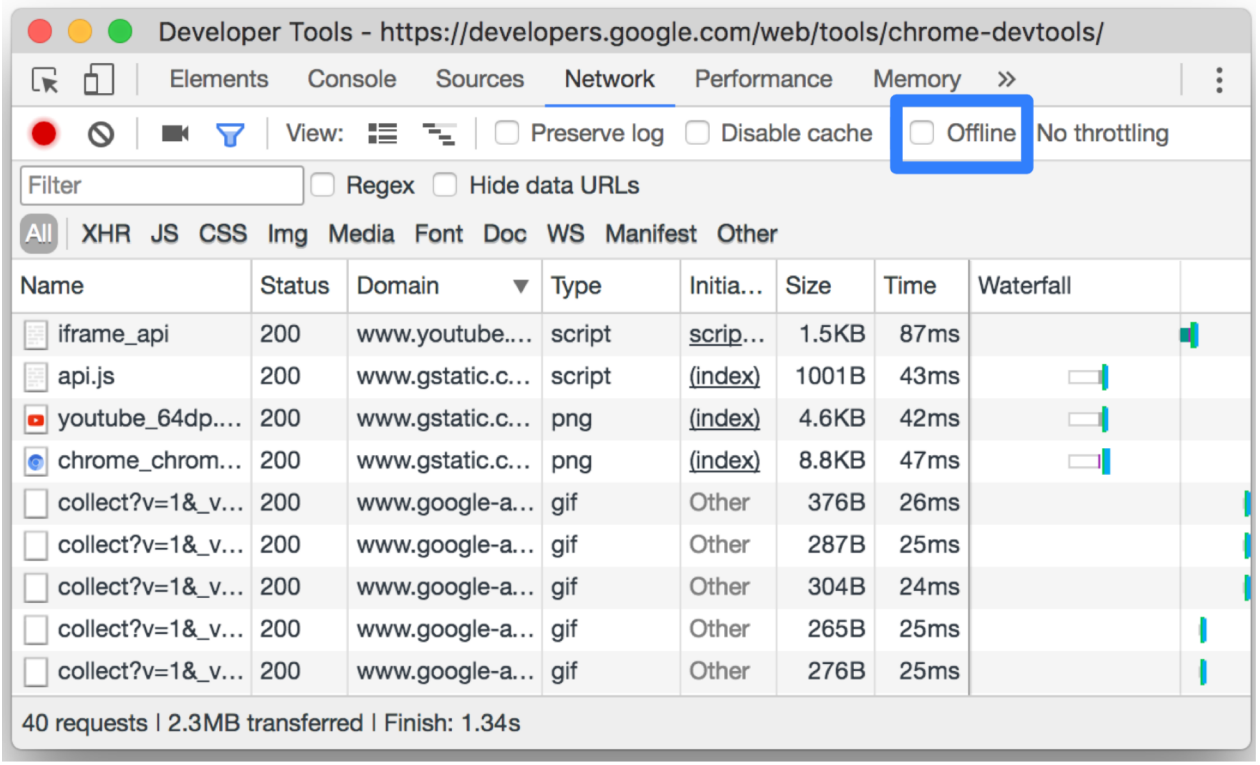


- You can also **manually clear the browser cache** at any time.
- Right-click anywhere in the **Requests** table
- Select **Clear Browser Cache**.



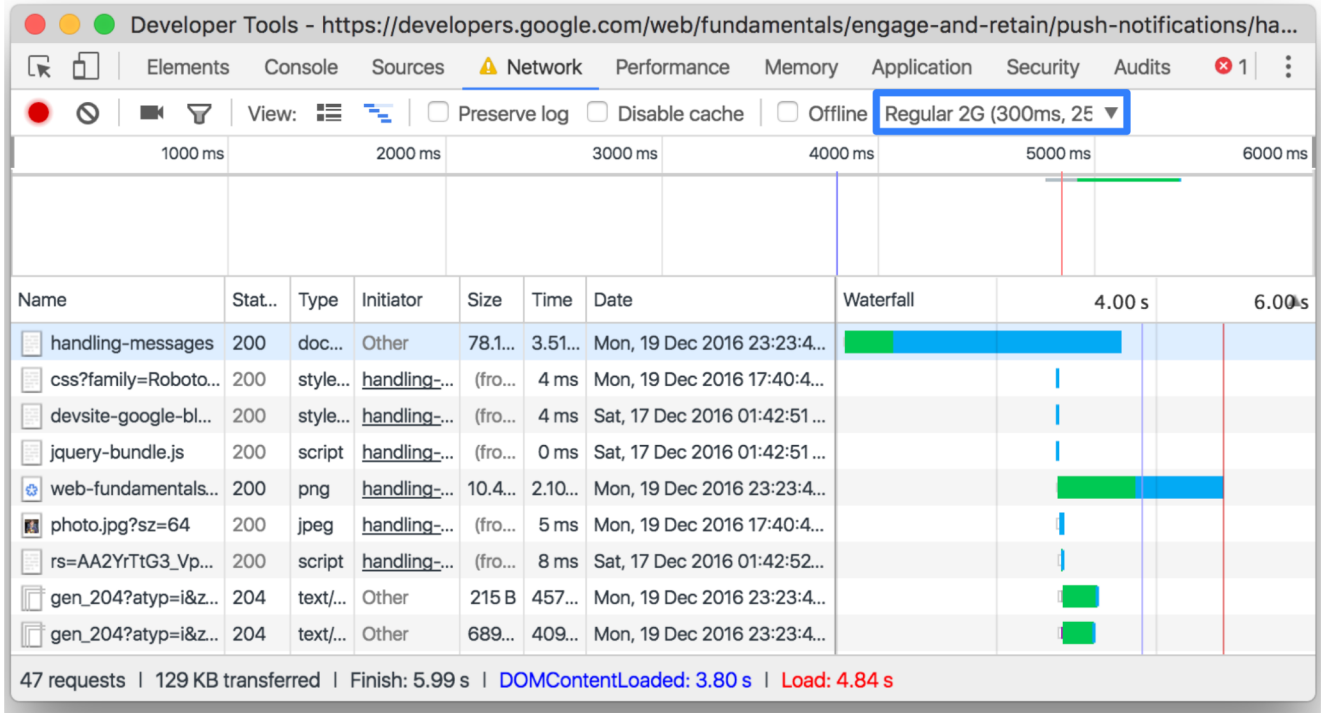
Emulate offline

- **Progressive Web Apps** are a kind of 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.



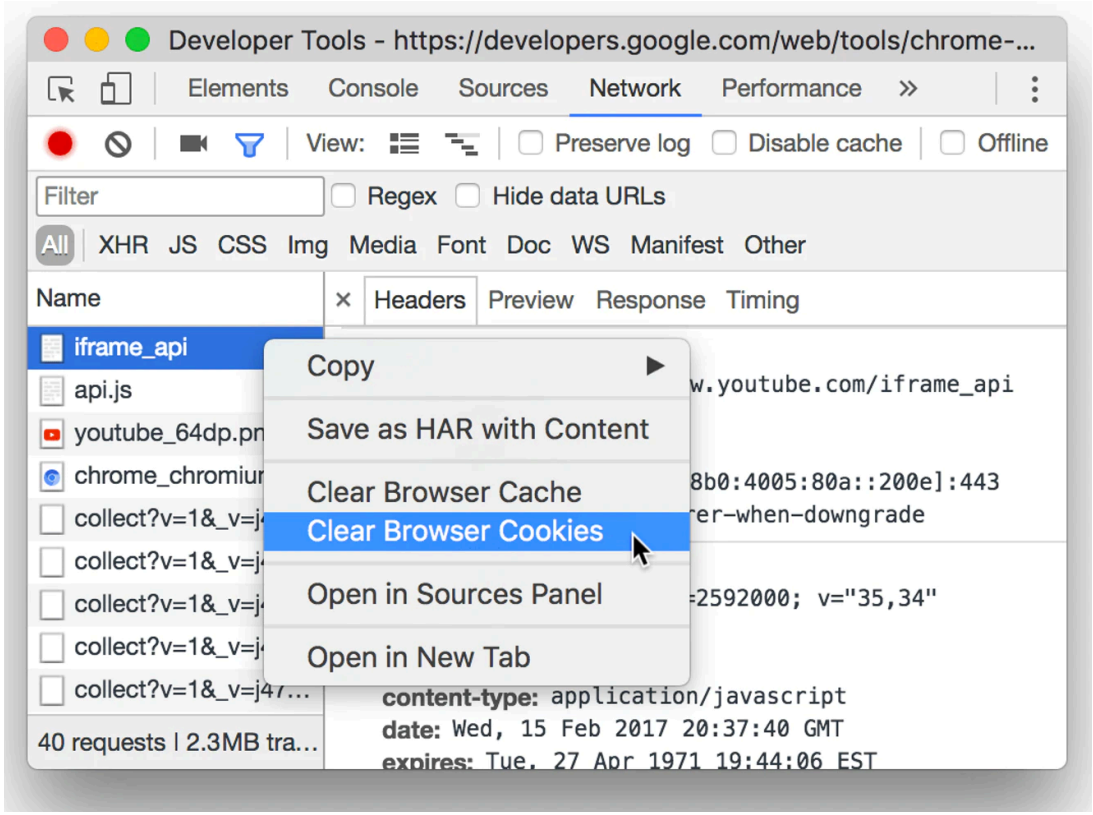
Emulate slow network connections

- Emulate 2G, 3G, and other connection speeds from the **Network Throttling** menu. You can select from a variety of presets.
- 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.



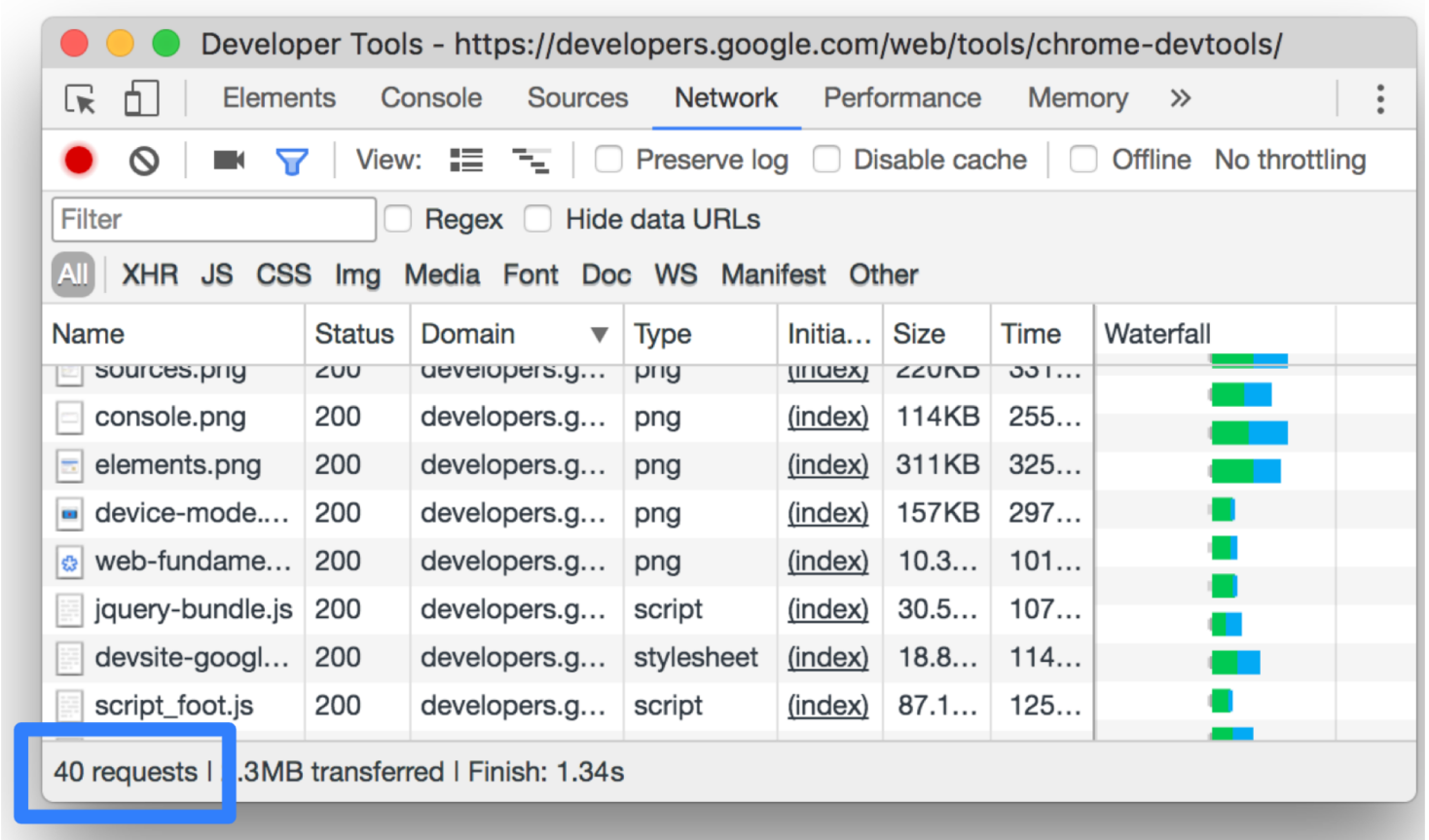
Clear browser cookies

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



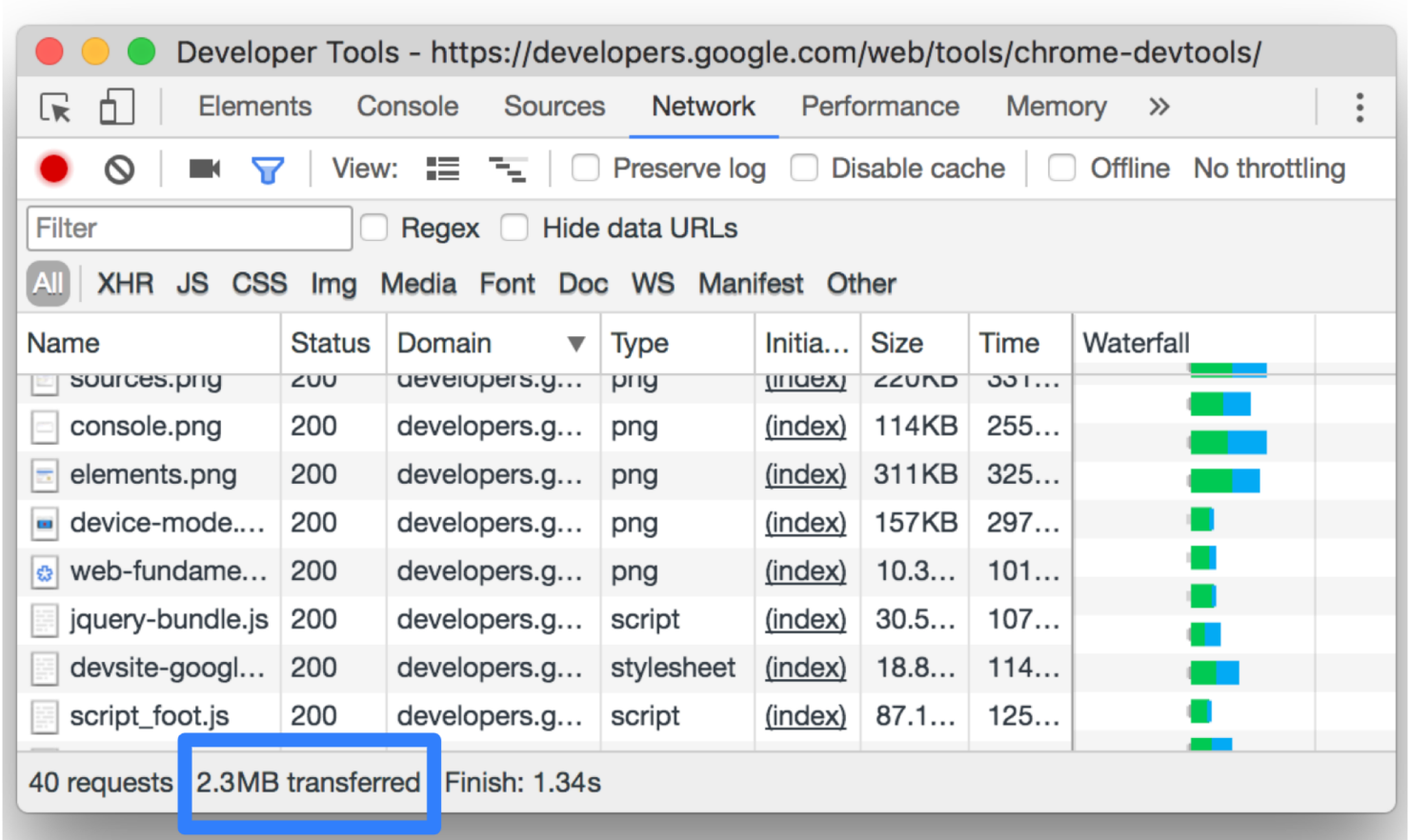
View the total number of requests

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



View the total download size

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



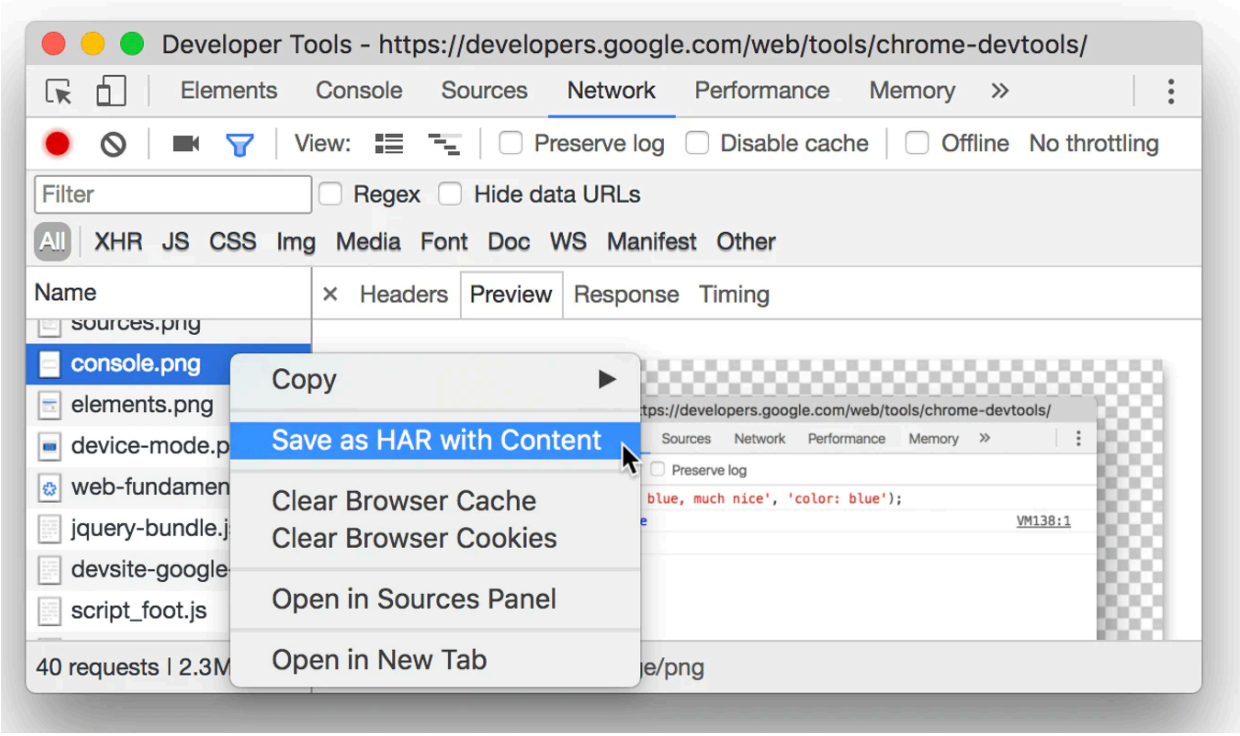
Export requests data

To save all network requests to a HAR file:

- Right-click any request in the Requests table.
- Select **Save as HAR with Content**.

DevTools saves all requests that have occurred since you opened DevTools to the HAR file.

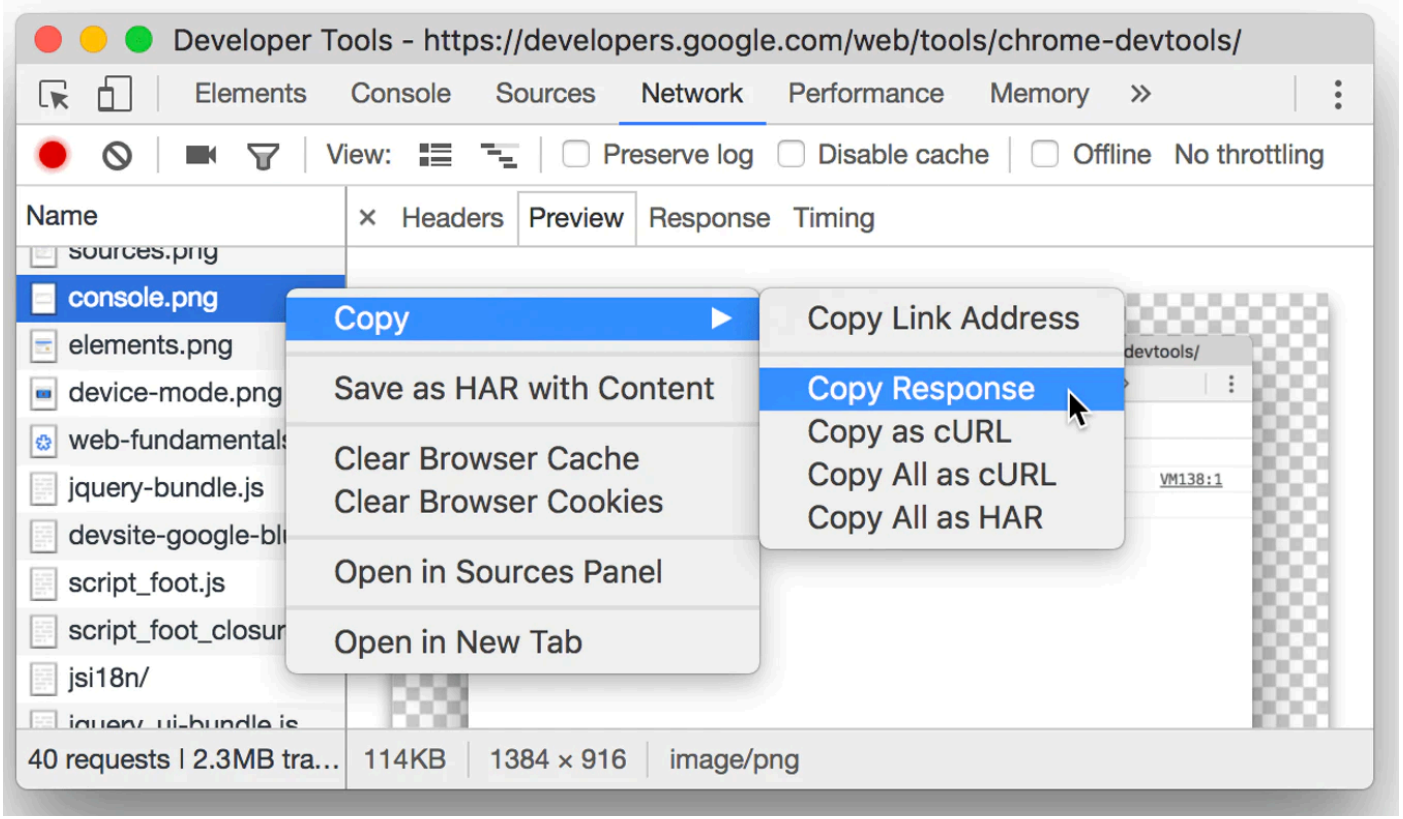
You can import a HAR file back into DevTools for analysis. Just drag-and-drop the HAR file into the Requests table.



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 Response**.
- **Copy as cURL**.
- **Copy All as cURL**.
- **Copy All as HAR**.



Assignment

Complete the **Inspect Network Activity** tutorial on Chrome DevTools website:

https://developer.chrome.com/docs/devtools/network/#open

Timeline Panel

Timeline Panel

- The **timeline** events mode displays all events triggered while making a recording.

- Use the timeline event reference to learn more about each timeline event type.

Common timeline event properties

Certain details are present in events of all types

Property	When it is shown
Aggregated time	For events with nested events, the time taken by each category of events.
Call Stack	For events with child events, the time taken by each category of events.
CPU time	How much CPU time the recorded event took.
Details	Other details about the event.
Duration (at time-stamp)	How long it took the event with all of its children to complete; timestamp is the time at which the event occurred, relative to when the recording started.
Self time	How long the event took without any of its children.
Used Heap Size	Amount of memory being used by the application when the event was recorded, and the delta (+/-) change in used heap size since the last sampling.

Loading events and properties

Event	Description
Parse HTML	Chrome executed its HTML parsing algorithm.
Finish Loading	A network request completed.
Receive Data	Data for a request was received. There will be one or more Receive Data events.
Receive Response	The initial HTTP response from a request.
Send Request	A network request has been sent.

Property	Description
Resource	The URL of the requested resource.
Preview	Preview of the requested resource (images only).
Request Method	HTTP method used for the request (GET or POST, for example).
Status Code	HTTP response code.
MIME Type	MIME type of the requested resource.
Encoded Data Length	Length of requested resource in bytes

Scripting events and properties

Event	Description
Event	A JavaScript event ("mousedown", or "key", for example).
GC Event	Garbage collection occurred.
DOMContentLoaded	The DOMContentLoaded was fired by the browser. This event is fired when all of the page's DOM content has been loaded and parsed.

Property	Description
Timer ID	The timer ID.
Timeout	The timeout specified by the timer.
Repeats	Boolean that specifies if the timer repeats.
Function Call	A function that was invoked.

Rendering events and properties

Event	Description
Invalidate layout	The page layout was invalidated by a DOM change.
Layout	A page layout was executed.
Recalculate style	Chrome recalculated element styles.
Scroll	The content of nested view was scrolled.

Property	Description
Layout Invalidated	The stack trace of the code that caused the layout to be invalidated.
Nodes that need layout	The number of nodes that were marked as needing layout before the relayout started. These are normally those nodes that were invalidated by developer code, plus a path upward to relayout root.
Layout tree size	The total number of nodes under the relayout root
Elements affected	The number of elements affected by a style recalculation.

Painting events and properties

Event	Description
Composite layers	Chrome's rendering engine composited image layers.
Image Decode	An image resource was decoded.
Image Resize	An image was resized from its native dimensions.
Paint	Composited layers were painted to a region of the display. Hovering over a Paint record highlights the region of the display that was updated

Property	Description
Location	For Paint events, the x and y coordinates of the paint rectangle.
Dimensions	For Paint events, the height and width of the painted region

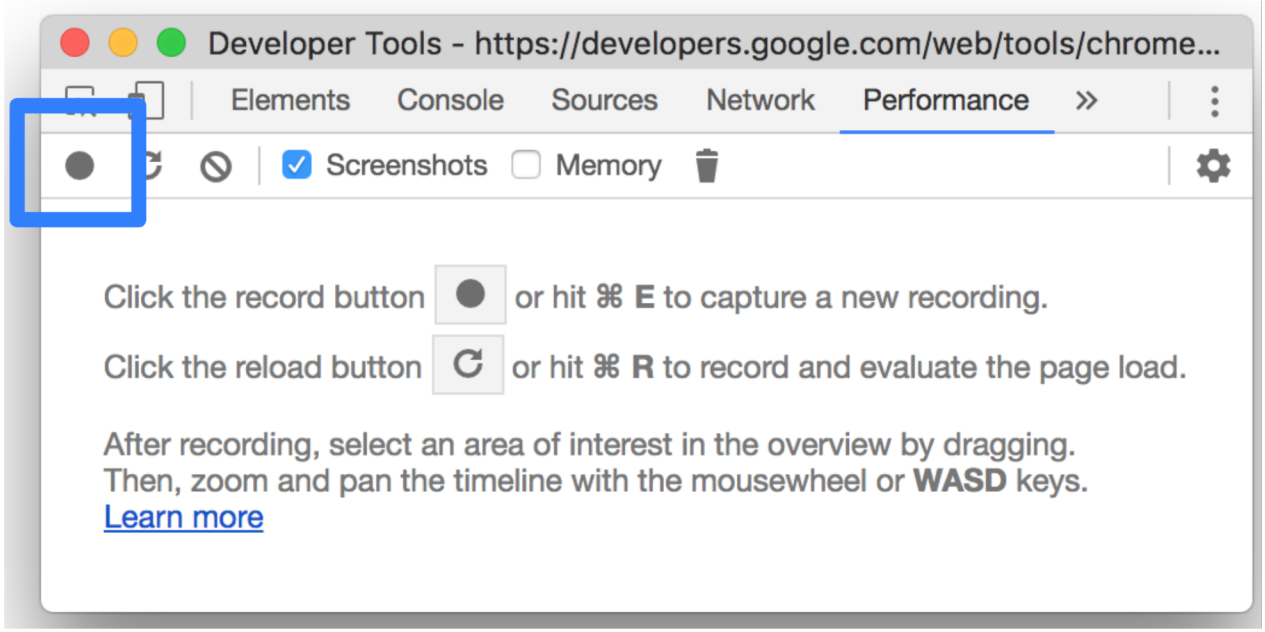
Profiles (Performance) Panel

Analyzing a page’s performance

Record runtime performance

Record runtime performance when you want to analyze the performance of a page as it's running.

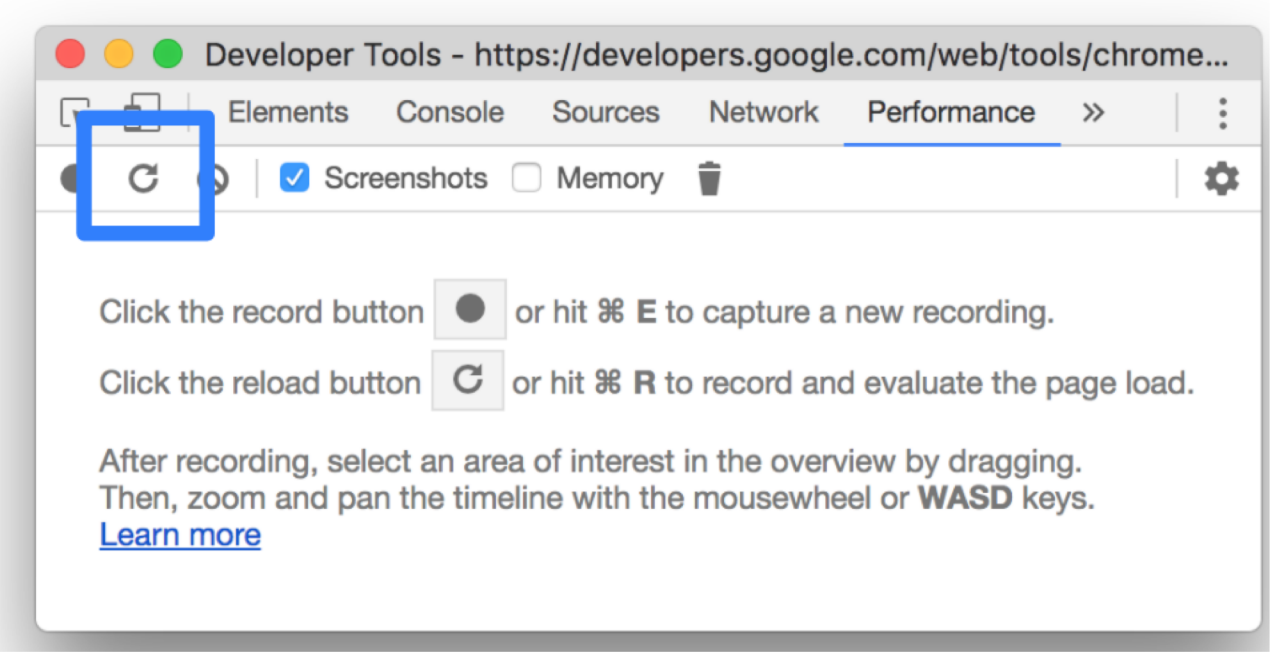
- Go to the page that you'll analyze.
- Click the **Performance** tab in DevTools.
- Click **Record**
- Interact with the page: DevTools records all page activity that occurs as a result of your interactions.
- Click **Record** again or click **Stop** to stop recording.



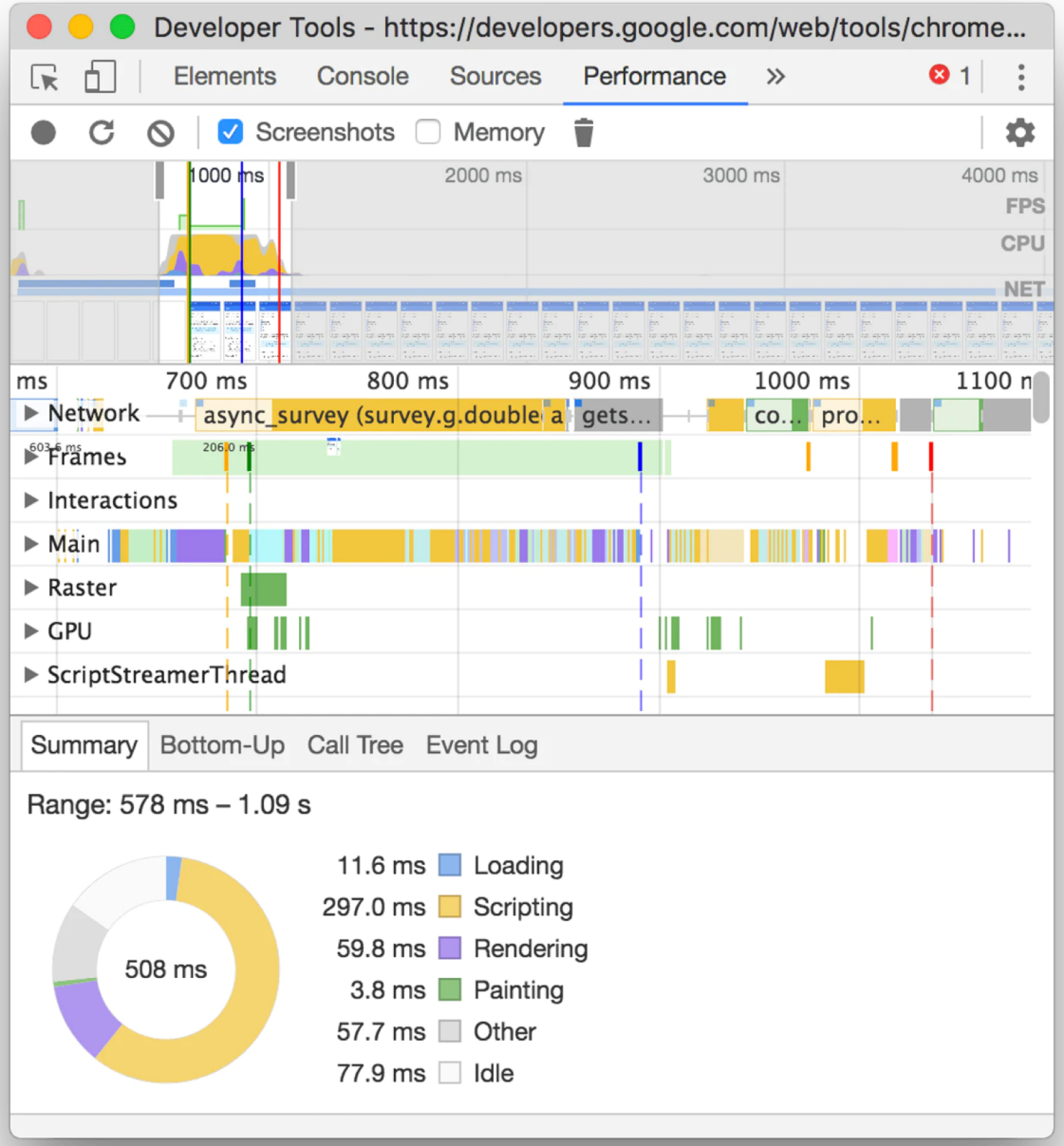
Record load performance

Record load performance when you want to analyze the performance of a page as it's loading.

- Go to the page that you'll analyze.
- Open the **Performance** panel of DevTools.
- Click **Reload page.**
- DevTools records performance metrics while the page reloads and then automatically stops the recording a couple seconds after the load finishes.

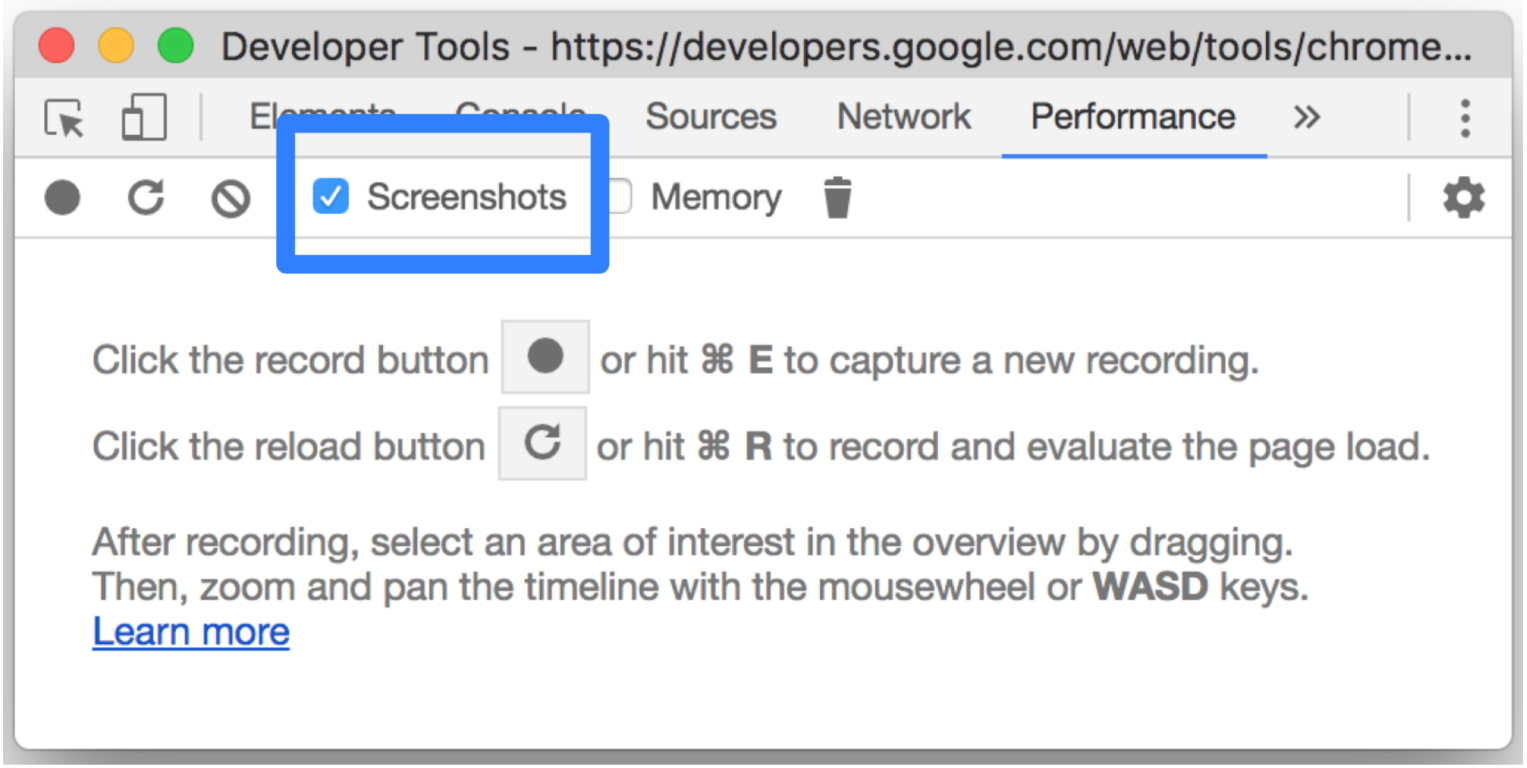


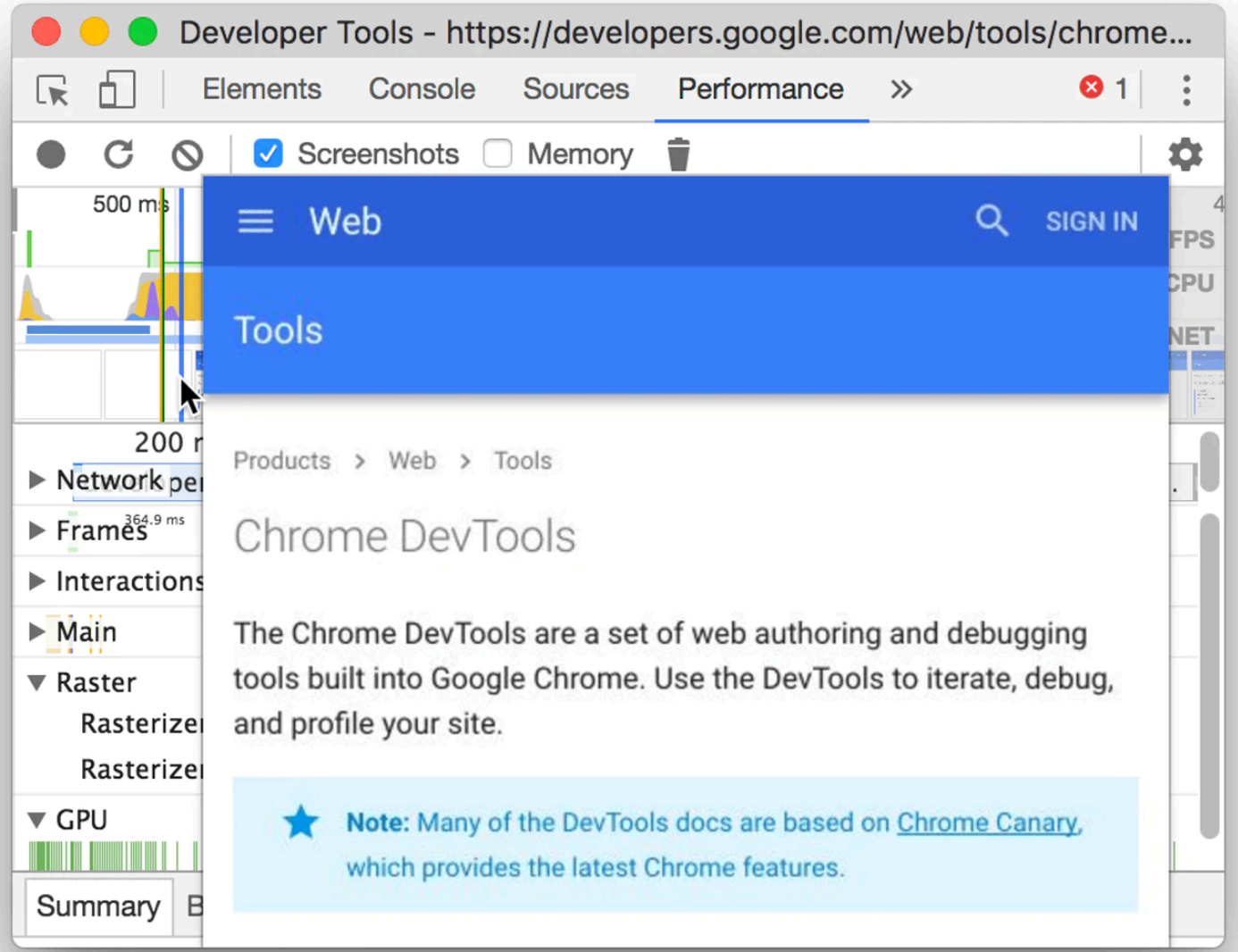
DevTools automatically zooms in on the portion of the recording where most of the activity occurred.



Capture screenshots while recording

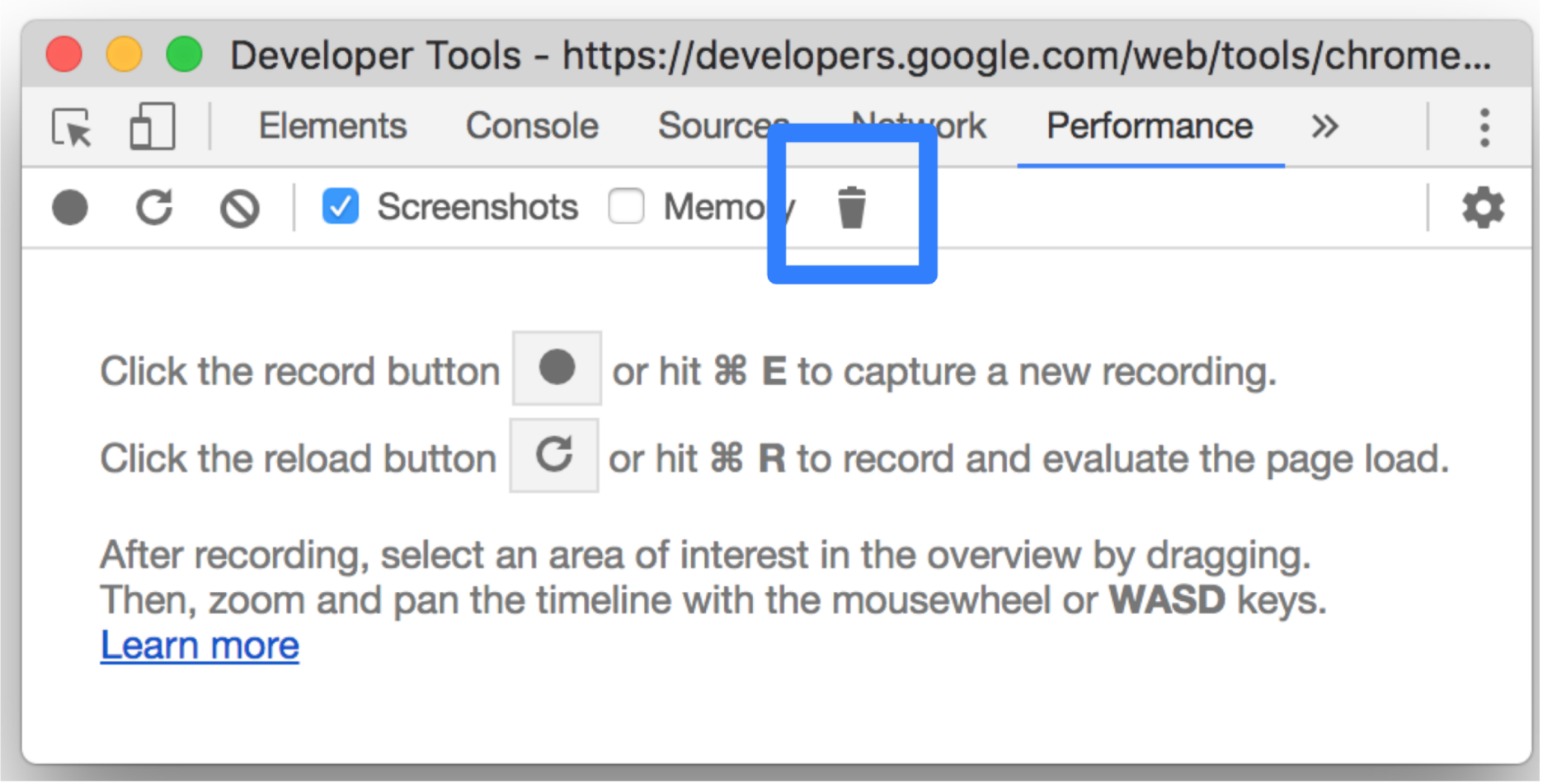
Enable the **Screenshots** checkbox to capture a screenshot of every frame while recording.




Developer Tools - https://developers.google.com/web/tools/chrome...
Elements
Console
Sources
Performance
1
Screenshots
Memory
500 ms
200 r
Network pe
Frames
364.9 ms
Interactions
Main
Raster
Rasterizer
Rasterizer
GPU
Summary
Web
SIGN IN
Tools
Products > Web > Tools
Chrome DevTools
The Chrome DevTools are a set of web authoring and debugging tools built into Google Chrome. Use the DevTools to iterate, debug, and profile your site.
Note: Many of the DevTools docs are based on Chrome Canary, which provides the latest Chrome features.
FPS
CPU
NET

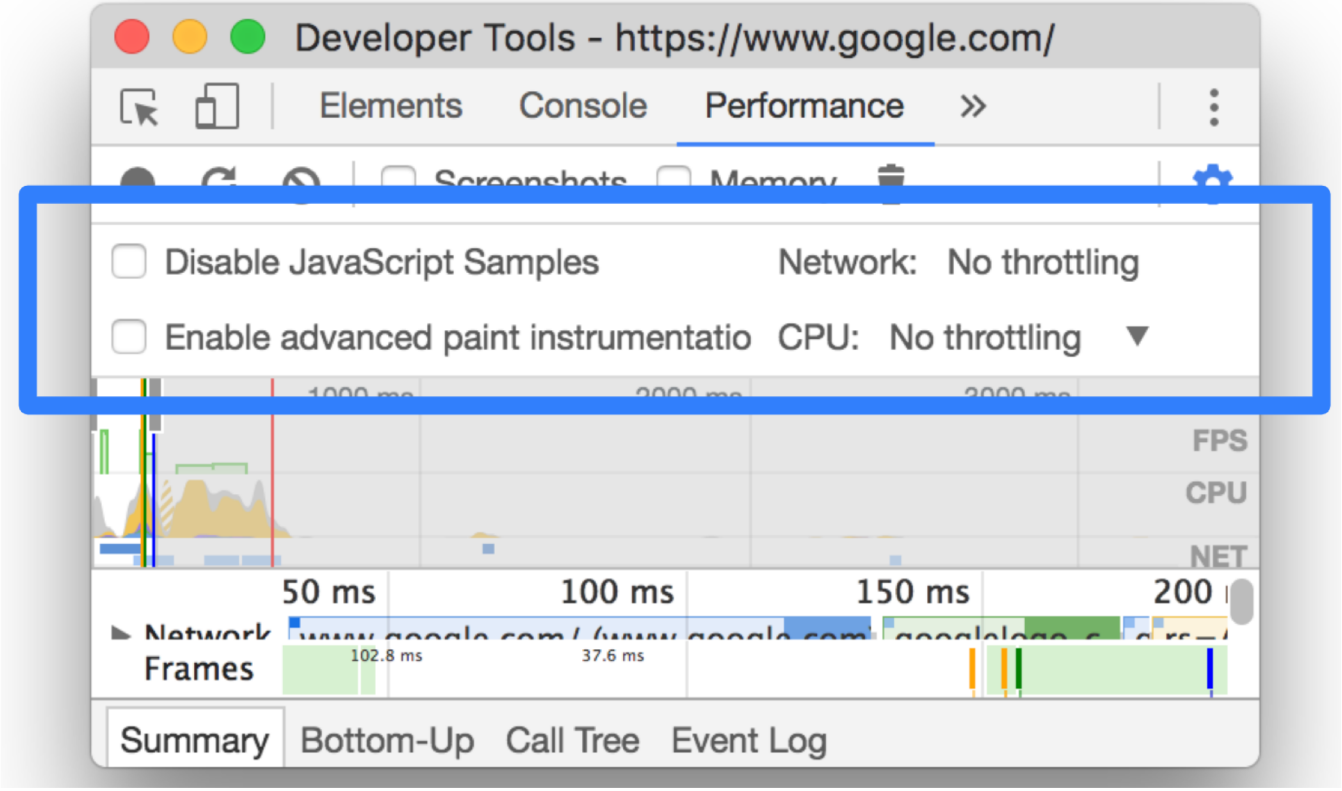
Force garbage collection while recording

While you are recording a page, click **Collect garbage** to force garbage collection.



Show recording settings

Click **Capture Settings** to expose more settings related to how DevTools captures performance recordings.



Assignment

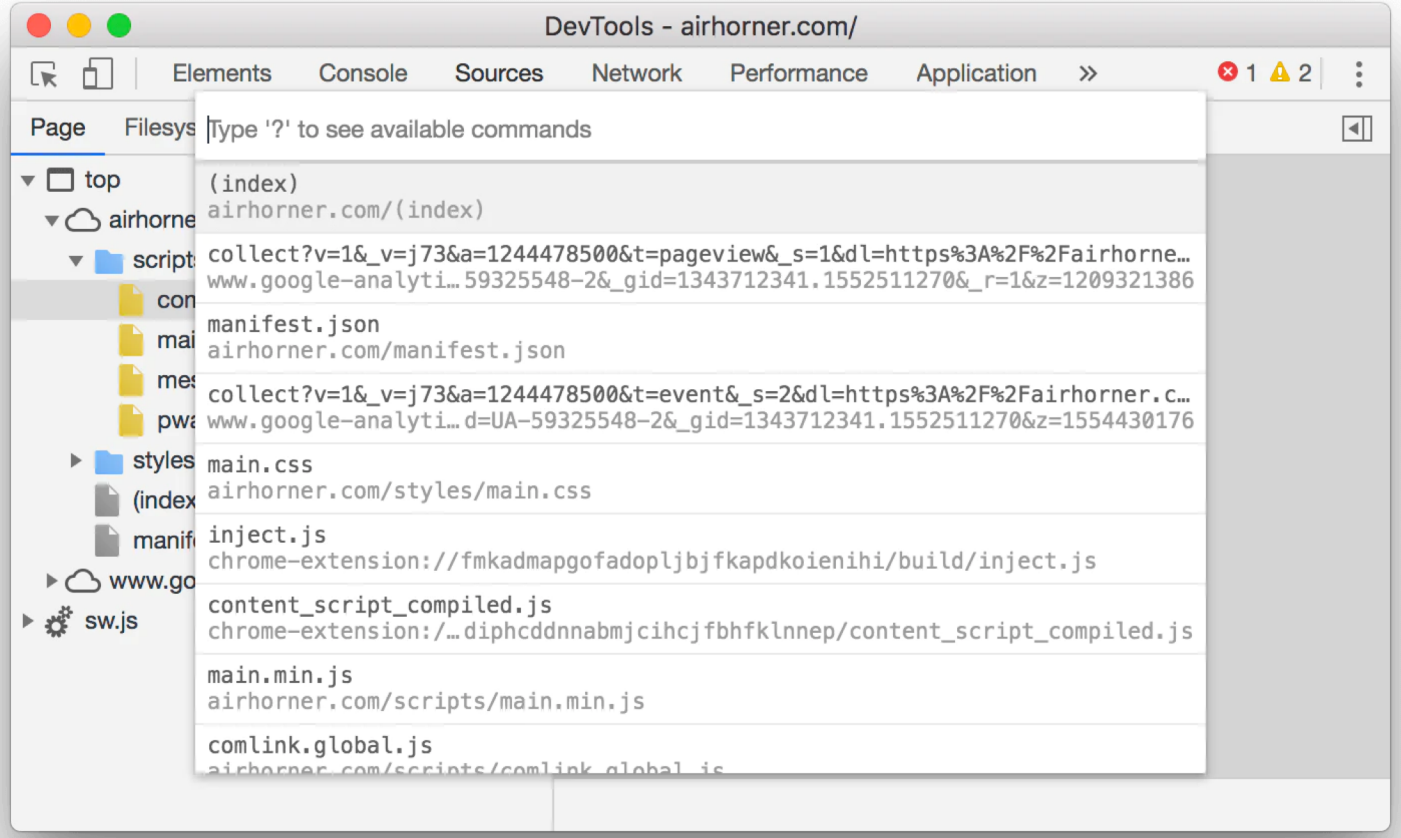
Complete the **Analyze runtime performance** tutorial on Chrome DevTools website:

https://developer.chrome.com/docs/devtools/evaluate-performance/

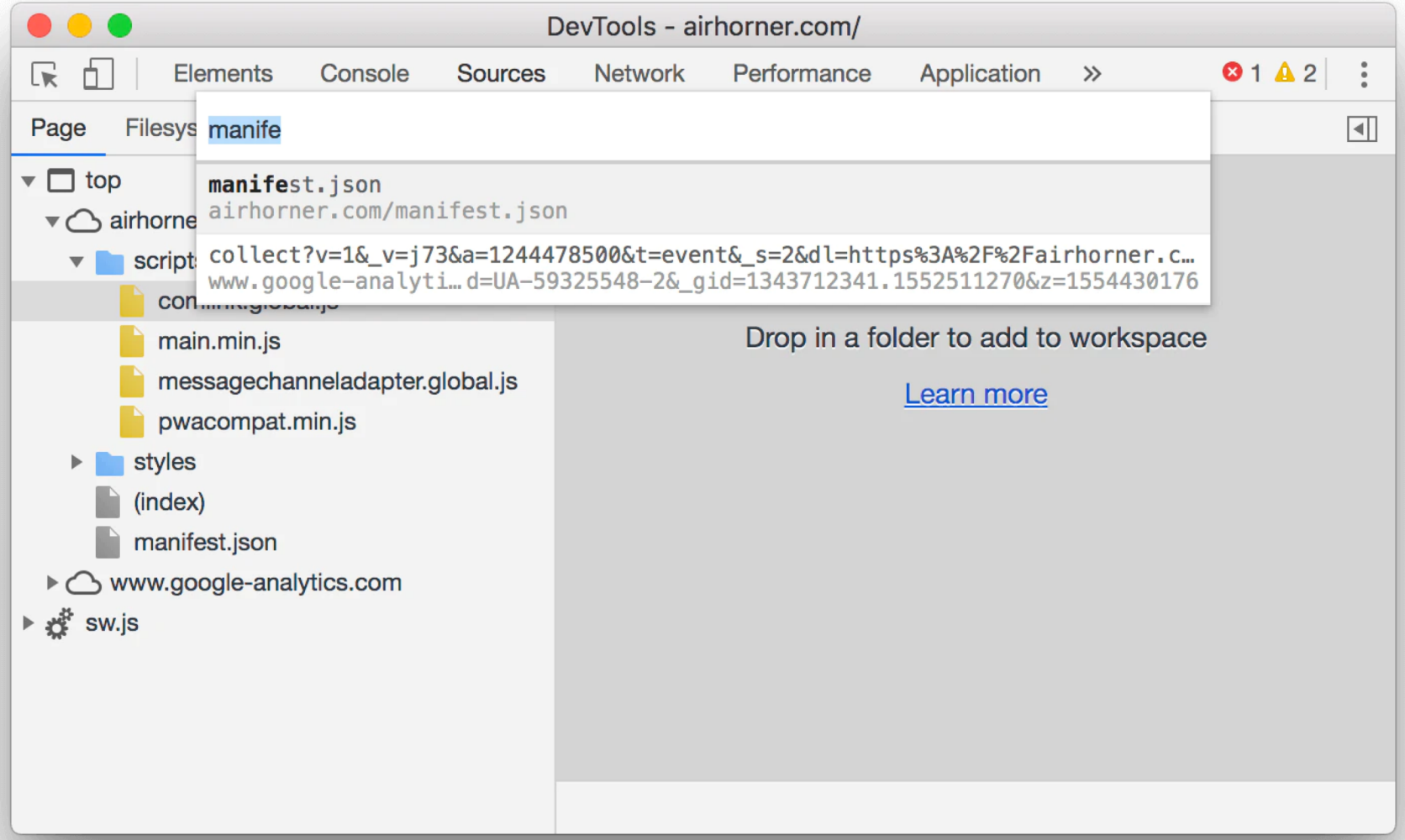
Resources Panel

Resources Panel

- Resources are the files that a page needs in order to display correctly.
- Examples of resources include CSS, JavaScript, and HTML files, as well as images.
- On Chrome DevTools, press **Control+P** or **Command+P** (Mac).
- The **Open File** dialog opens.



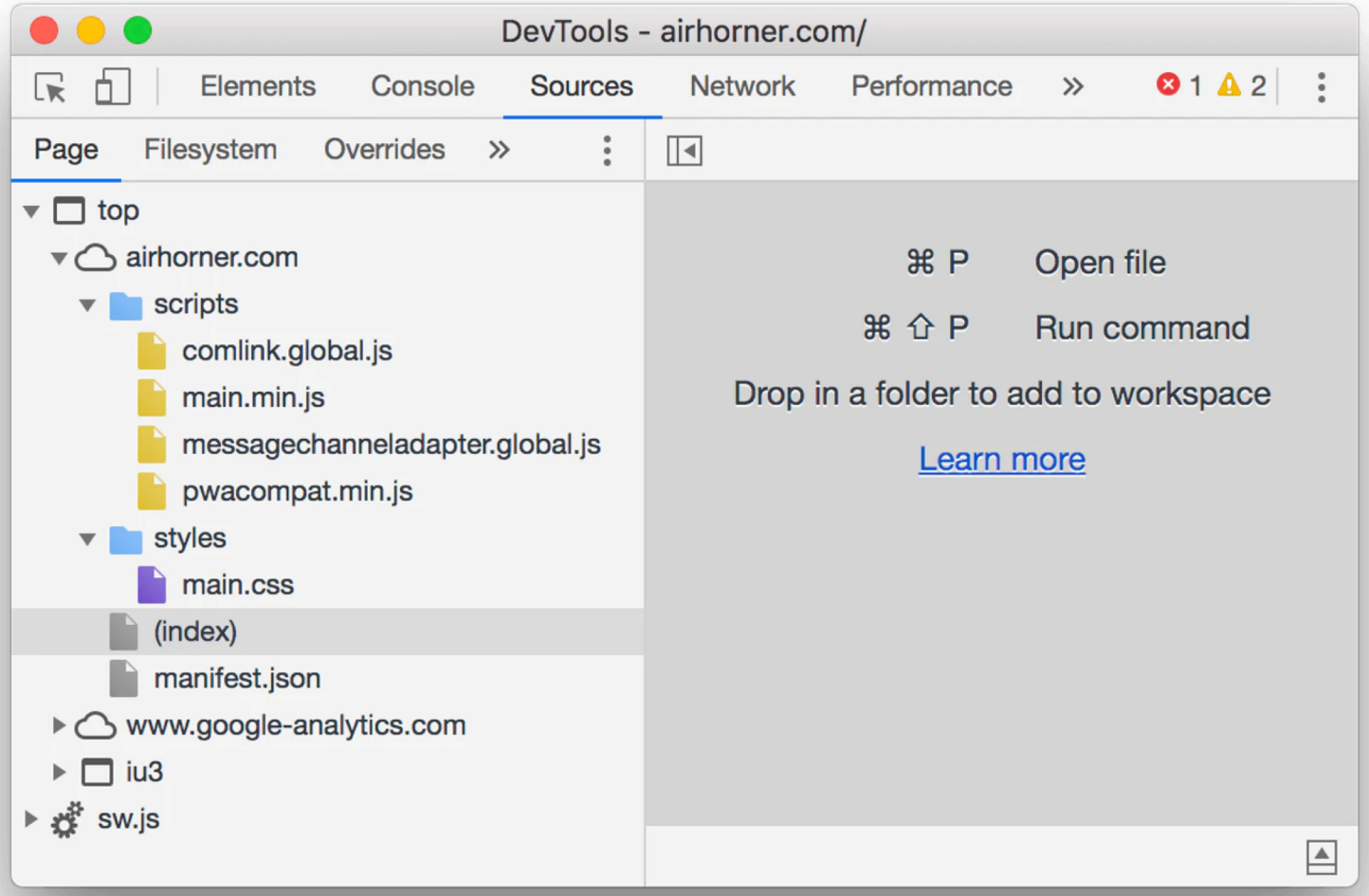
Select the file from the dropdown, or start typing the filename and press Enter once the correct file is highlighted in the autocomplete box.



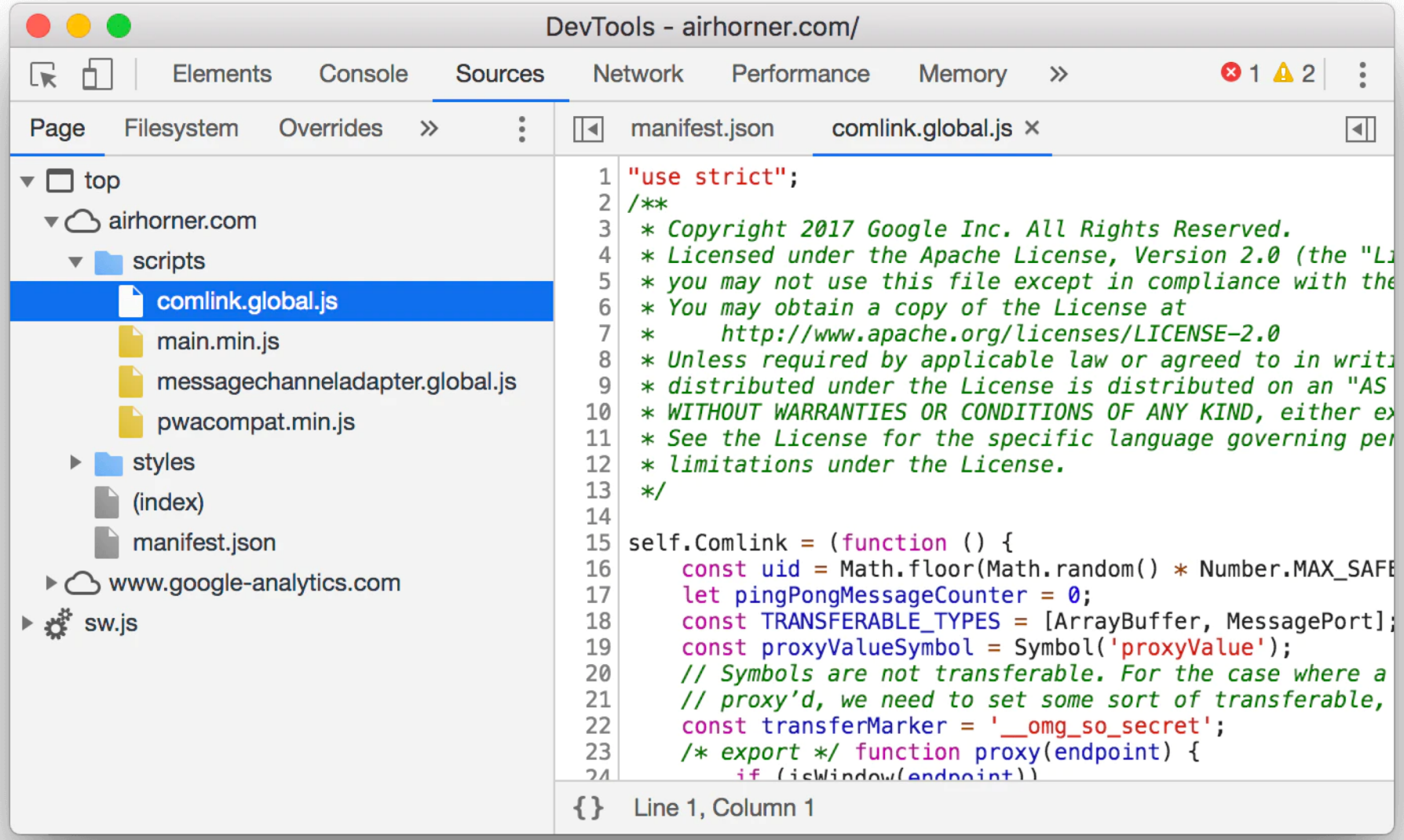
Browse by directory

To view a page's resources organized by directory:

- Click the **Sources** tab to open the **Sources** panel.
- Click the **Page** tab to show the page's resources. The **Page** pane opens.



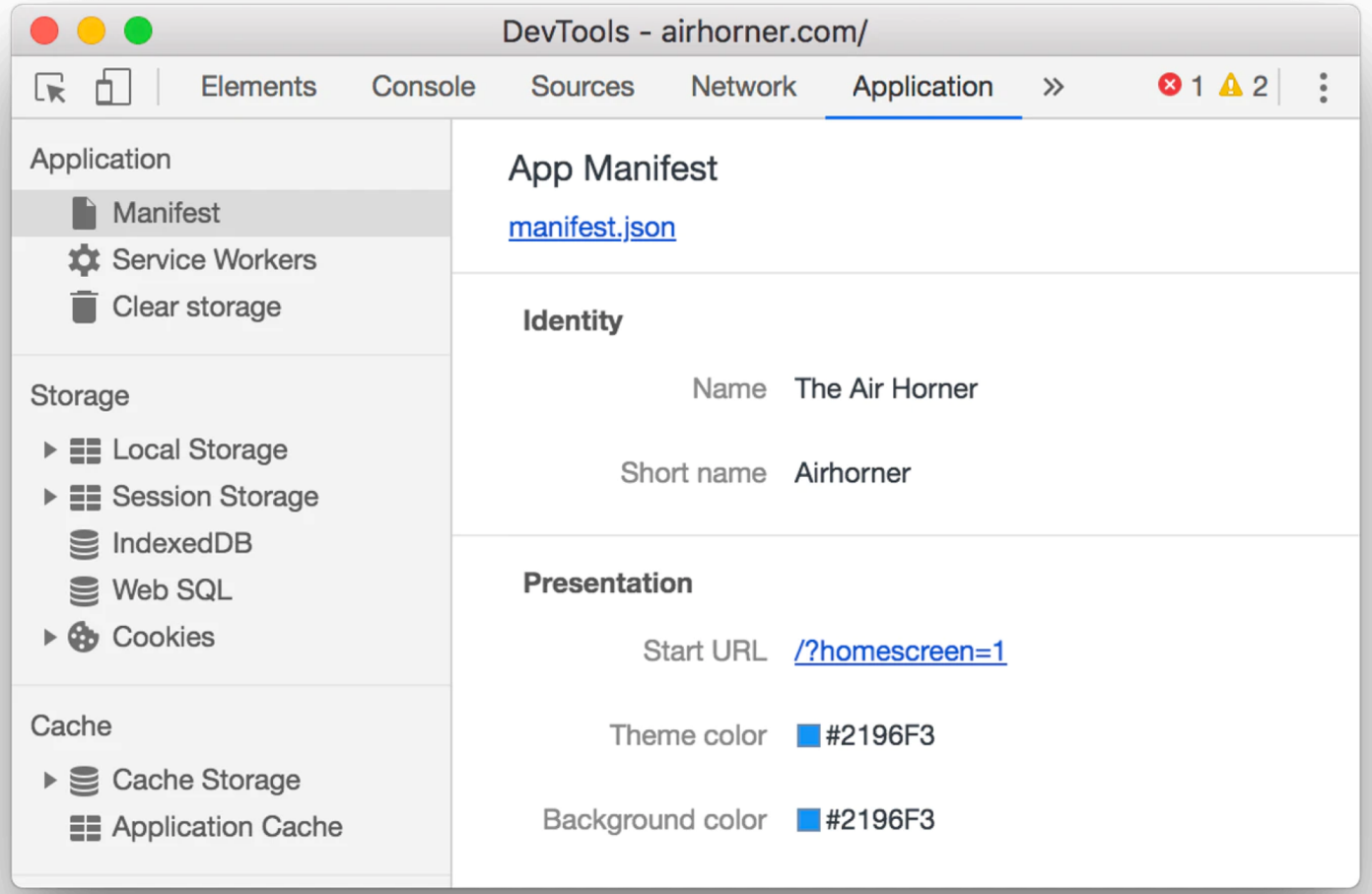
Click a resource to view it in the Editor



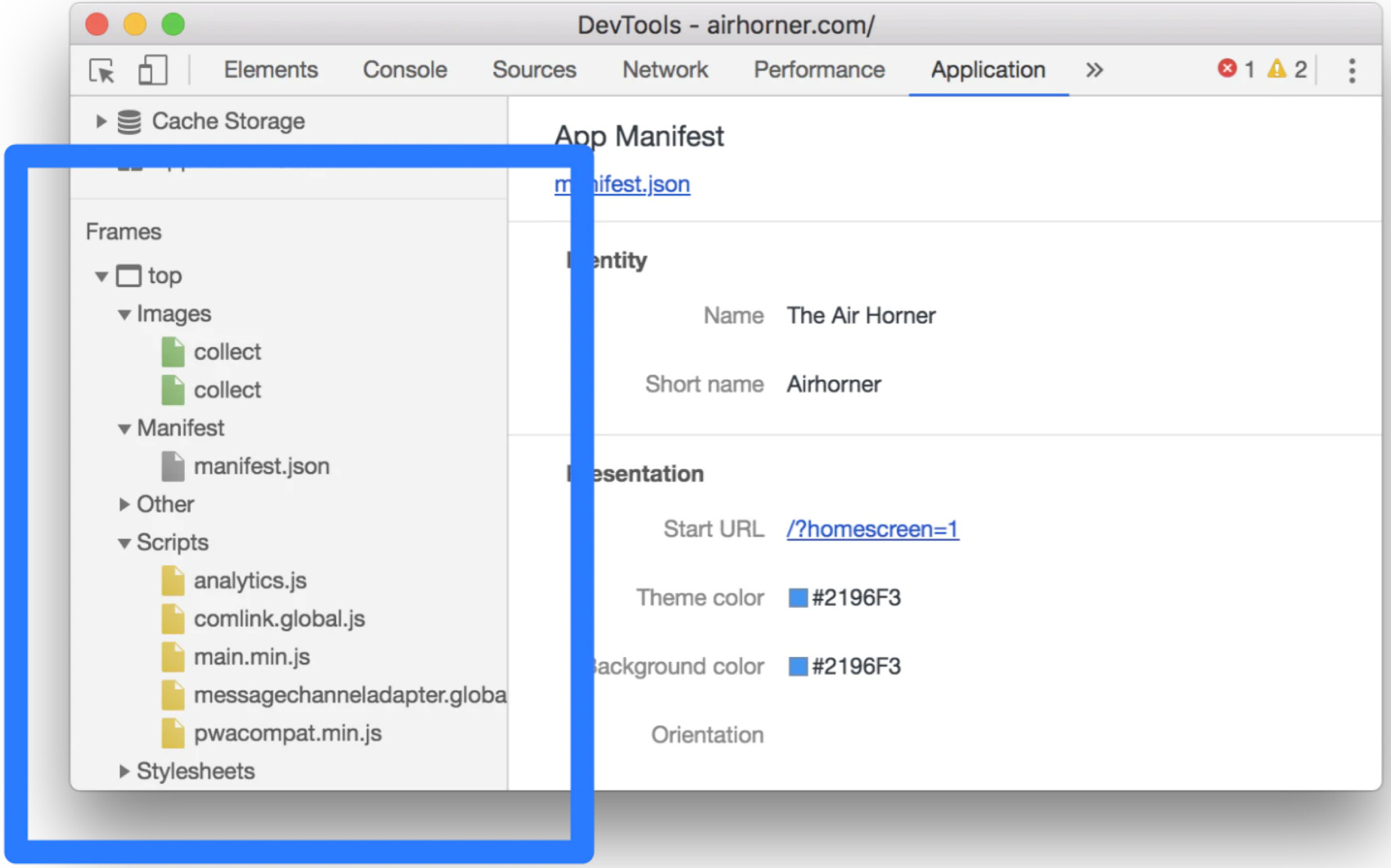
Browse by file type

To group resources together based on their file type:

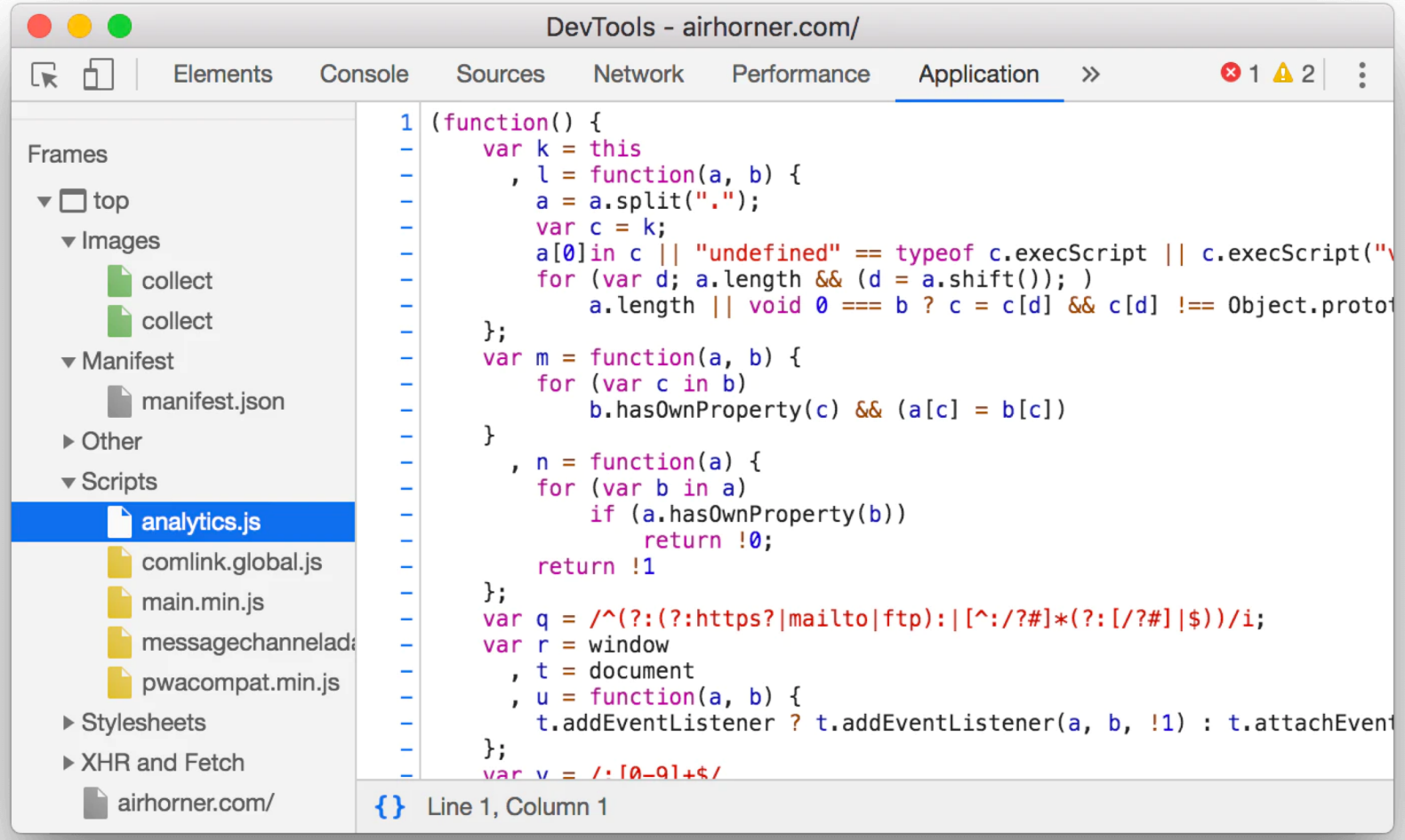
- Click the **Application** tab.
- The **Application** panel opens.
- By default the **Manifest** pane usually opens first.



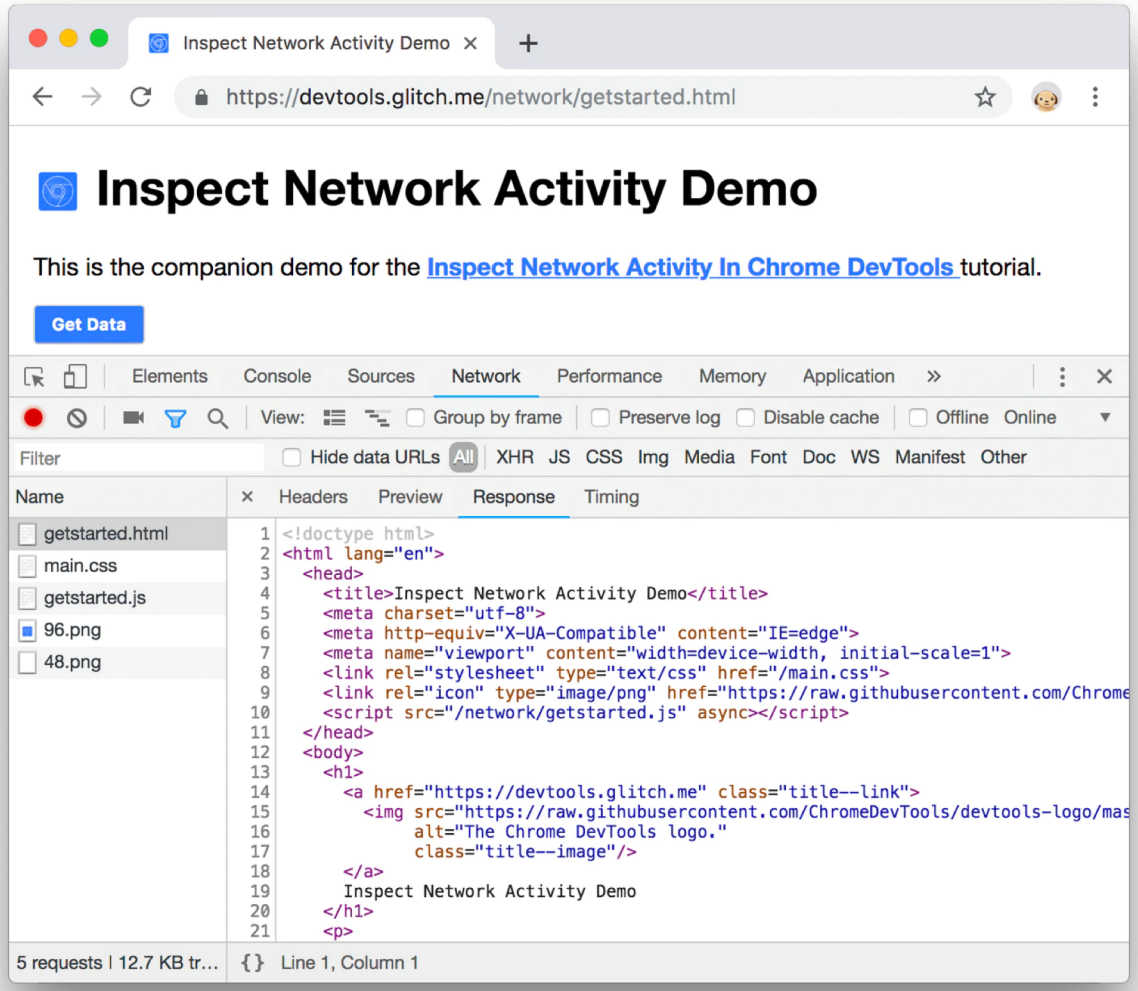
- Scroll down to the **Frames** pane.



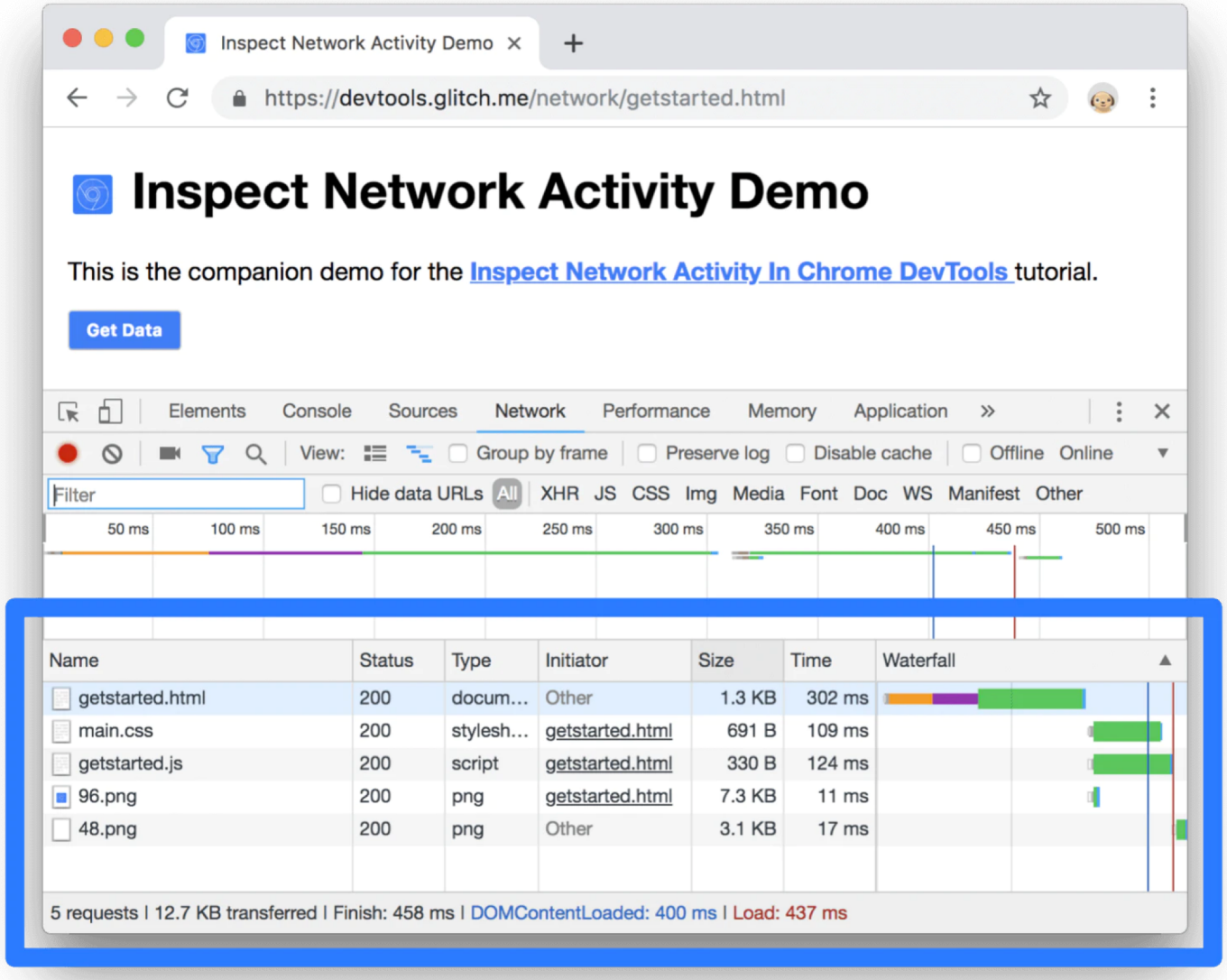
- Expand the sections that you're interested in.
- Click a resource to view it.



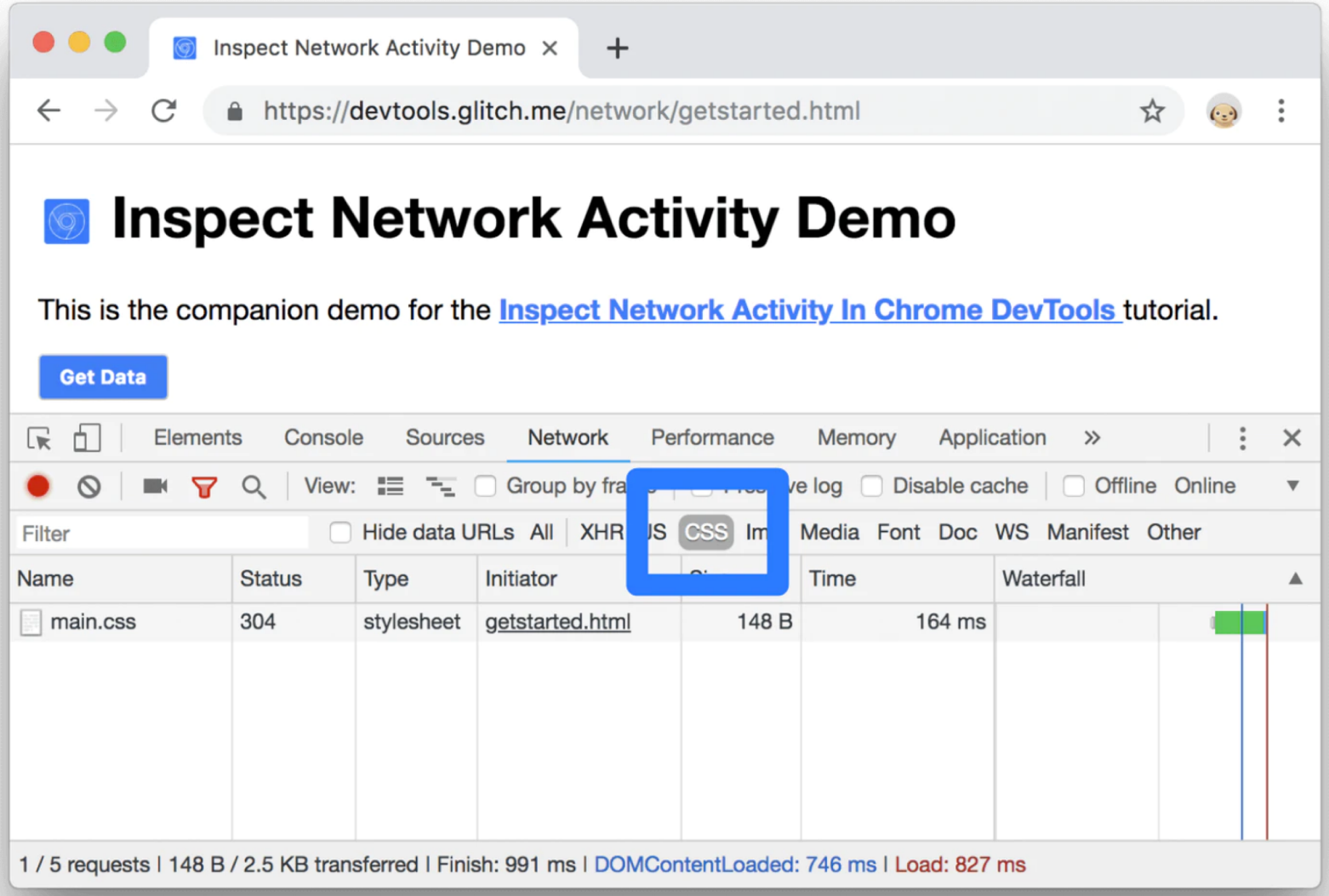
Open resources in the Network panel



Browse resources in the Network panel



Browse files by type in the Network panel

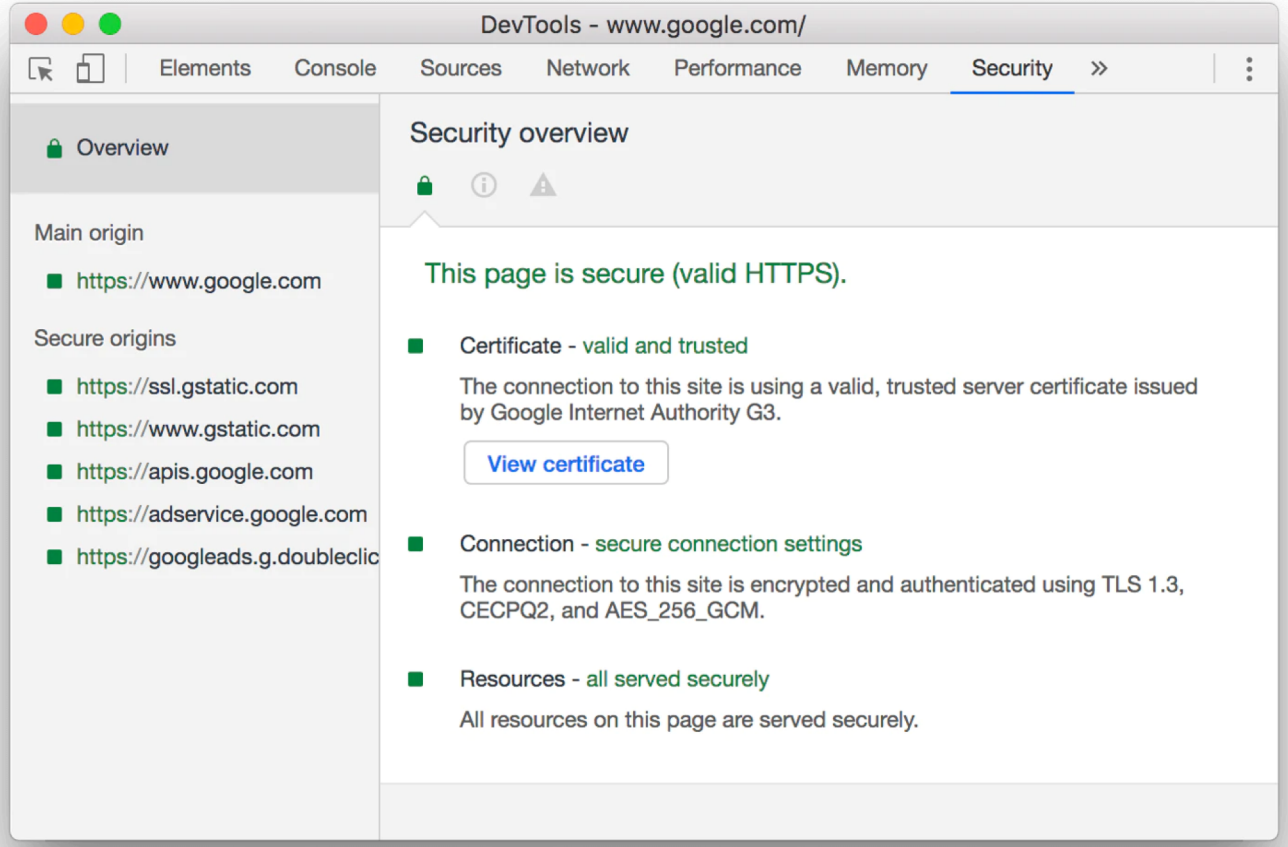


Security Panel

Making sure HTTPS is properly implemented on a page

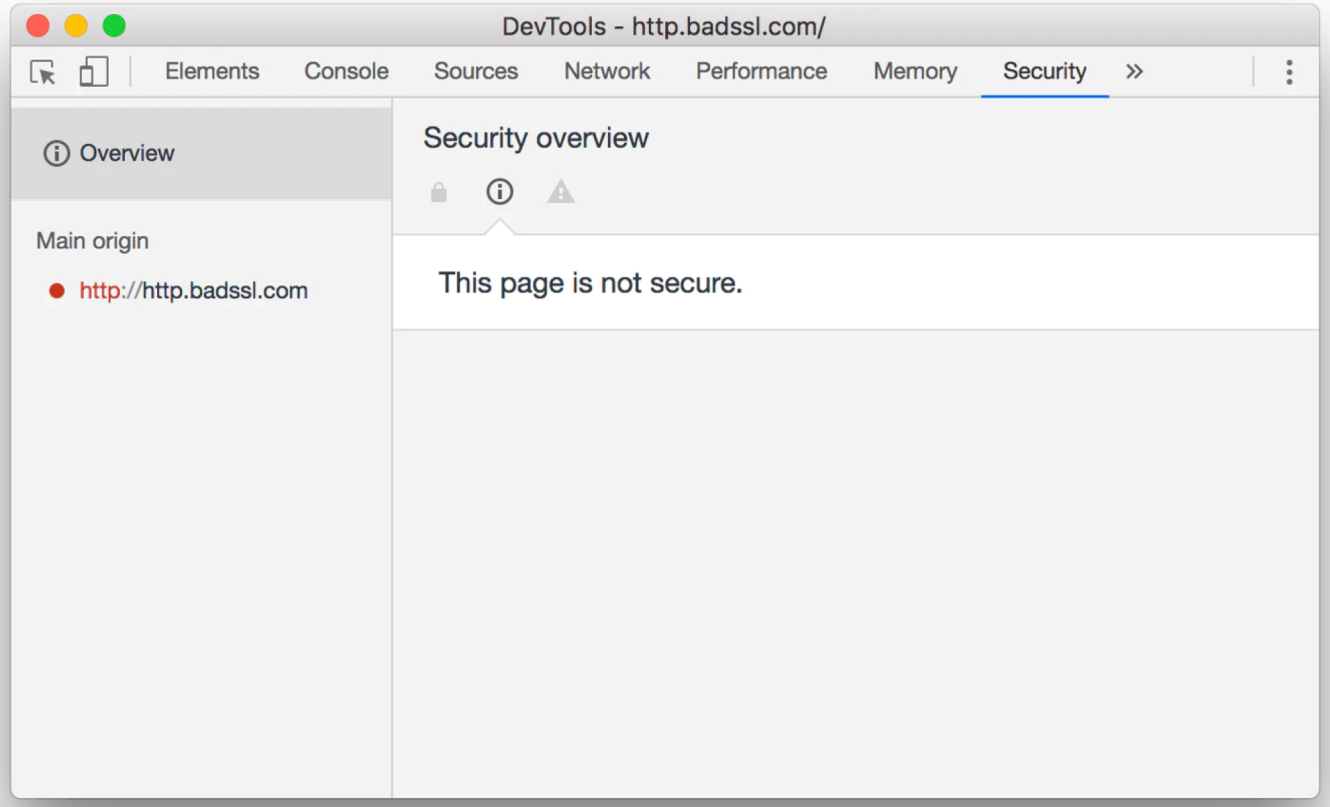
Security Panel

- The **Security** panel is the main place in DevTools for inspecting the security of a page.



Common problems: Non-secure main origins

- When the main origin of a page is not secure, the **Security Overview** says **This page is not secure**.
- This problem occurs when the URL that you visited was requested **over HTTP**.
- To make it secure you need to request it **over HTTPS and see if it works**.



- If you've already got HTTPS set up on your server, all you need to do to fix this problem is **configure your server to redirect all HTTP requests to HTTPS**.

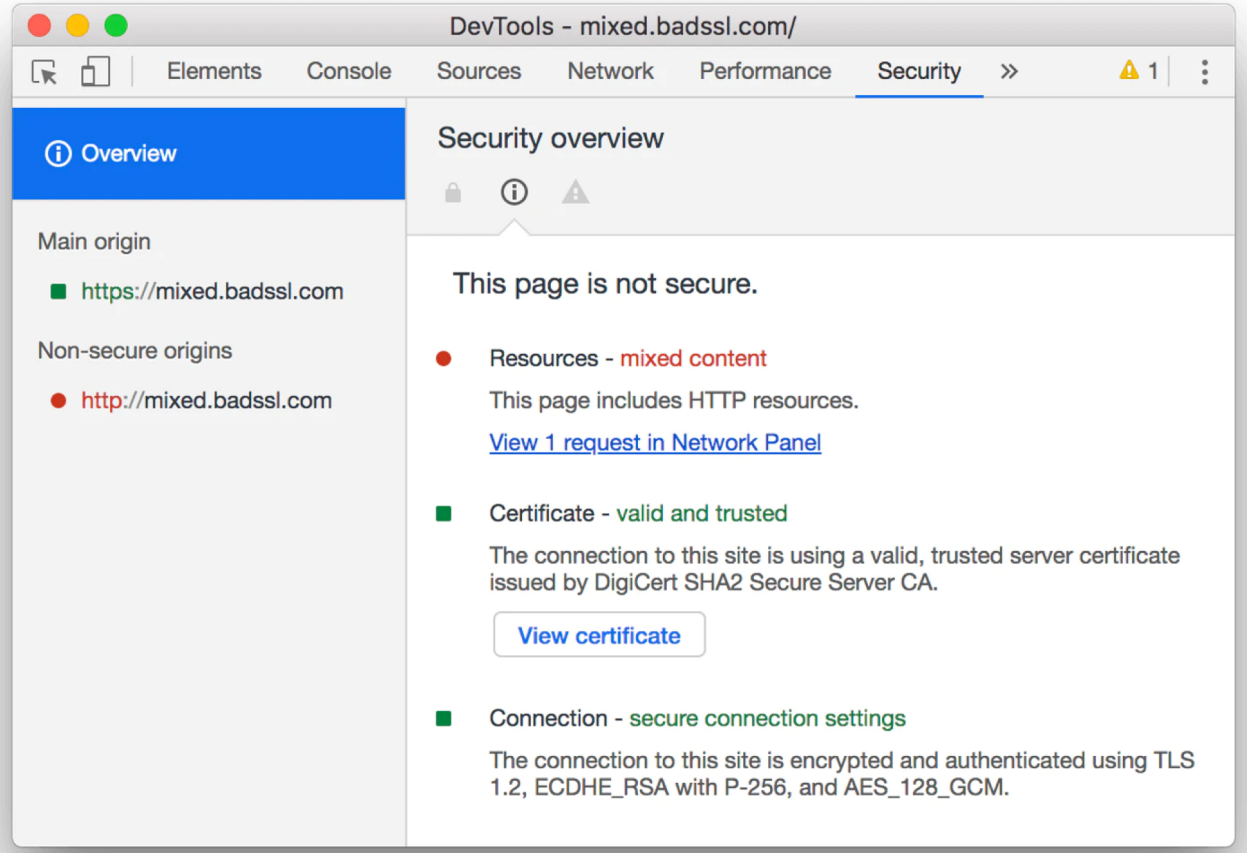
- If you don't have HTTPS set up on your server, **Let's Encrypt** provides a free and relatively-easy way to start the process.

- The **Redirect HTTP Traffic To HTTPS audit** can help automate the process of making sure that all HTTP requests are redirected to HTTPS.

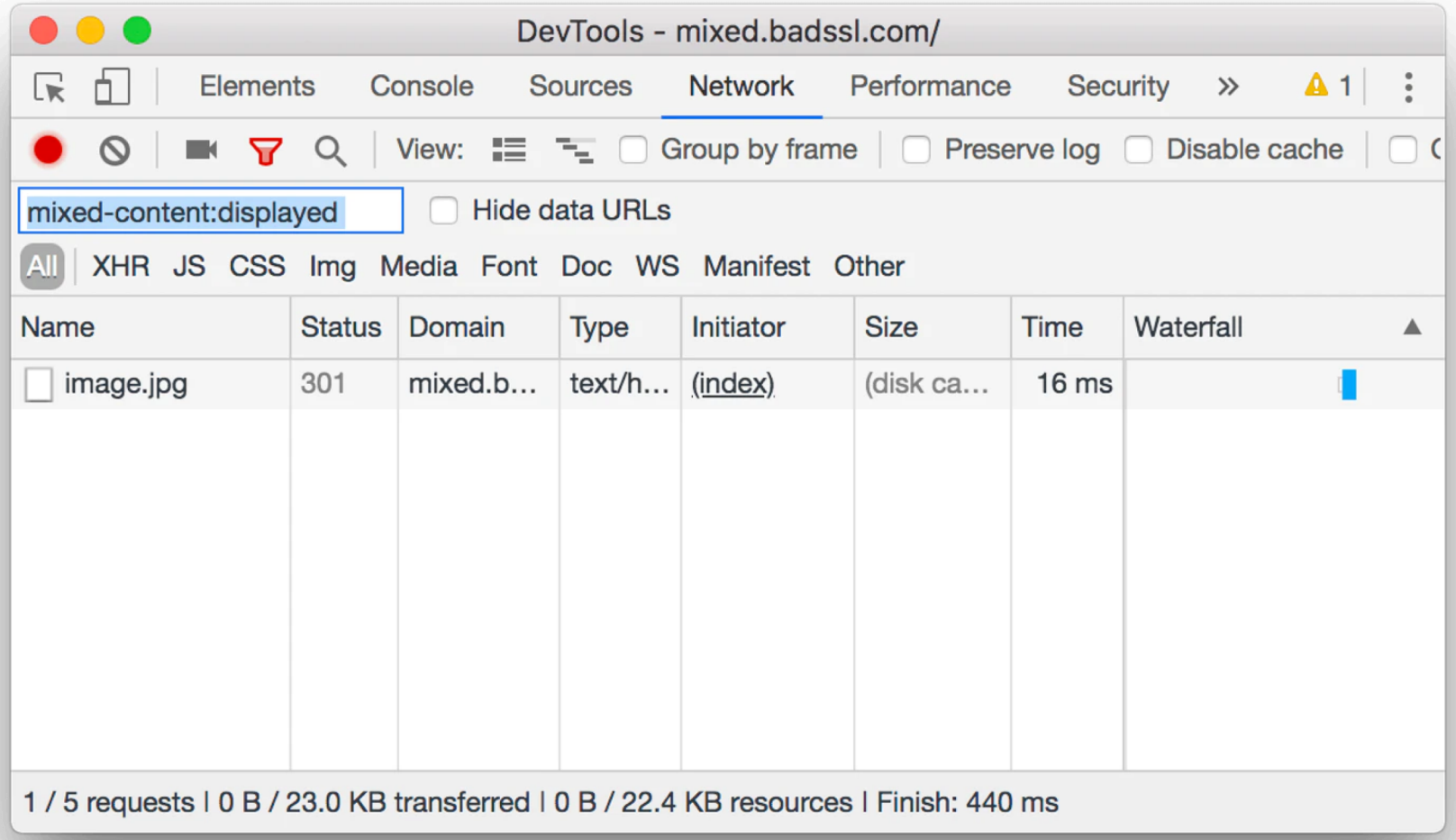
Common problems: Mixed content

- Mixed content means that the **main origin of a page is secure**, but the page **requested resources from non-secure origins**.

- Mixed content pages are only partially protected because the HTTP content is accessible to sniffers and vulnerable to man-in-the-middle attacks.

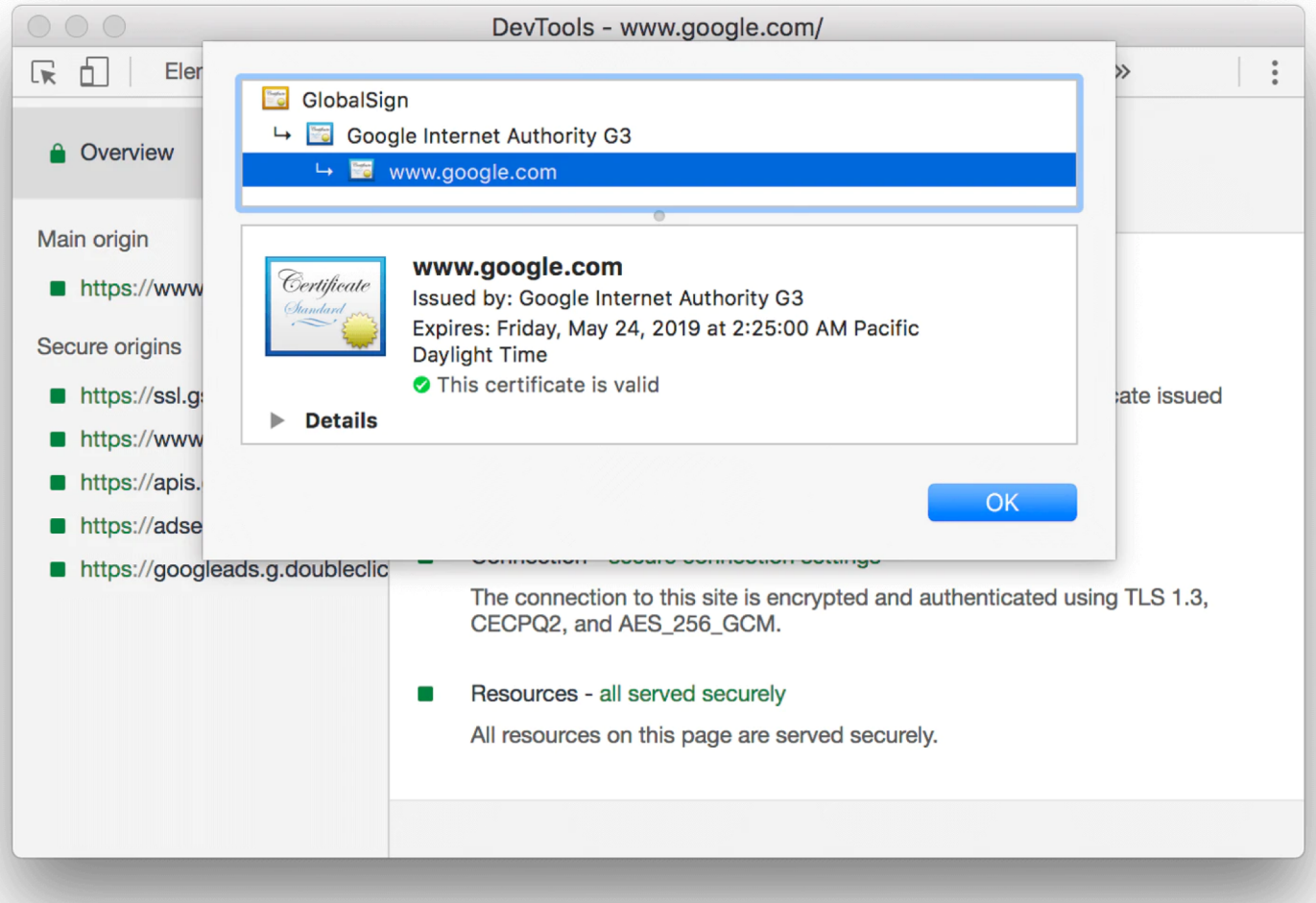


Clicking **View 1 request in Network panel** opens the **Network** panel and applies the **mixed-content:displayed** filter so that the **Network Log** only shows non-secure resources.



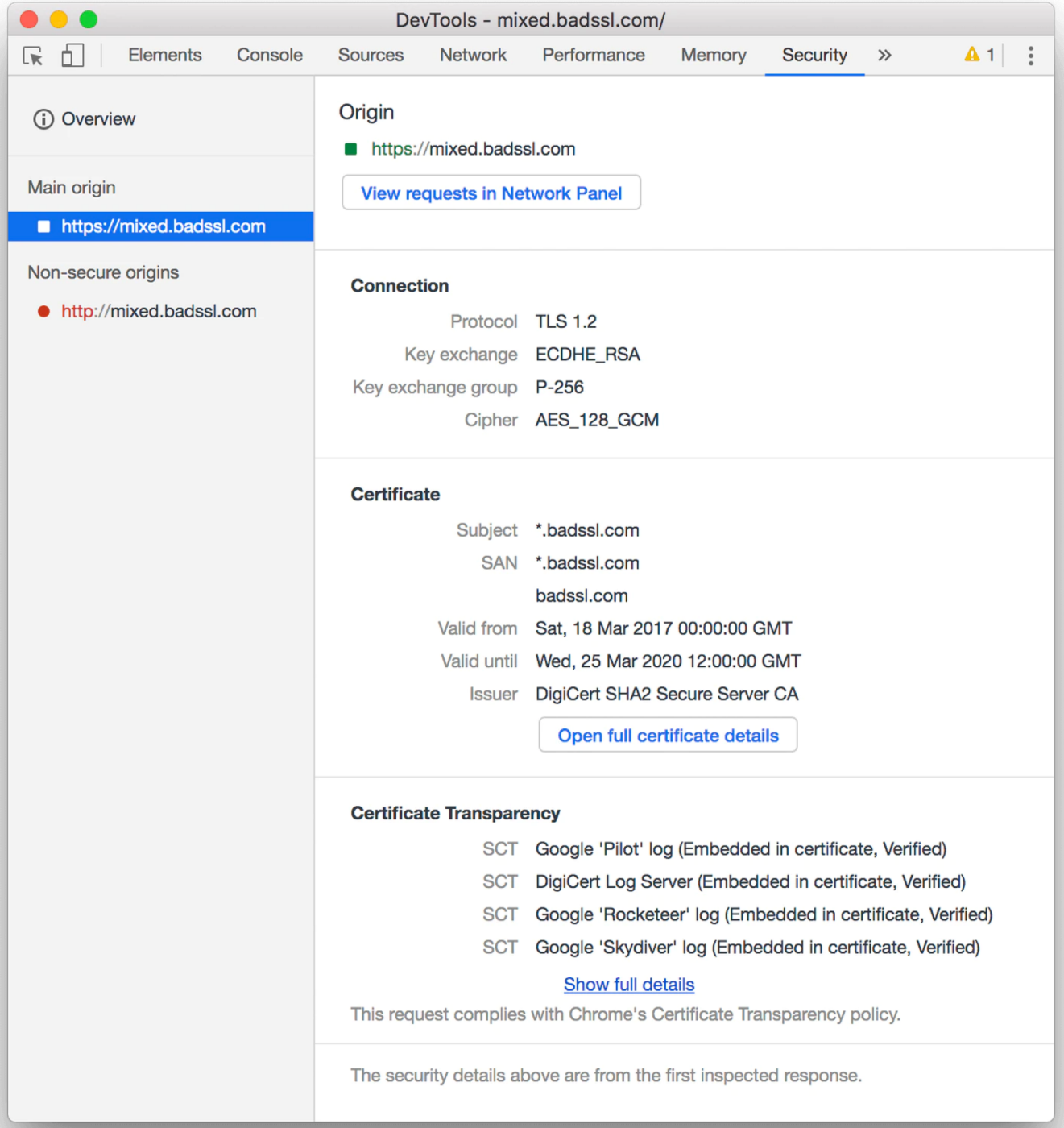
View details

From the **Security Overview** click **View certificate** to quickly inspect the main origin's certificate.



View origin details

- Click one of the entries in the left-hand nav to view the origin's details.
- From the details page you can view **connection** and **certificate** information.
- Certificate transparency information is also shown when available.



Audits (Lighthouse) Panel

Audits Panel

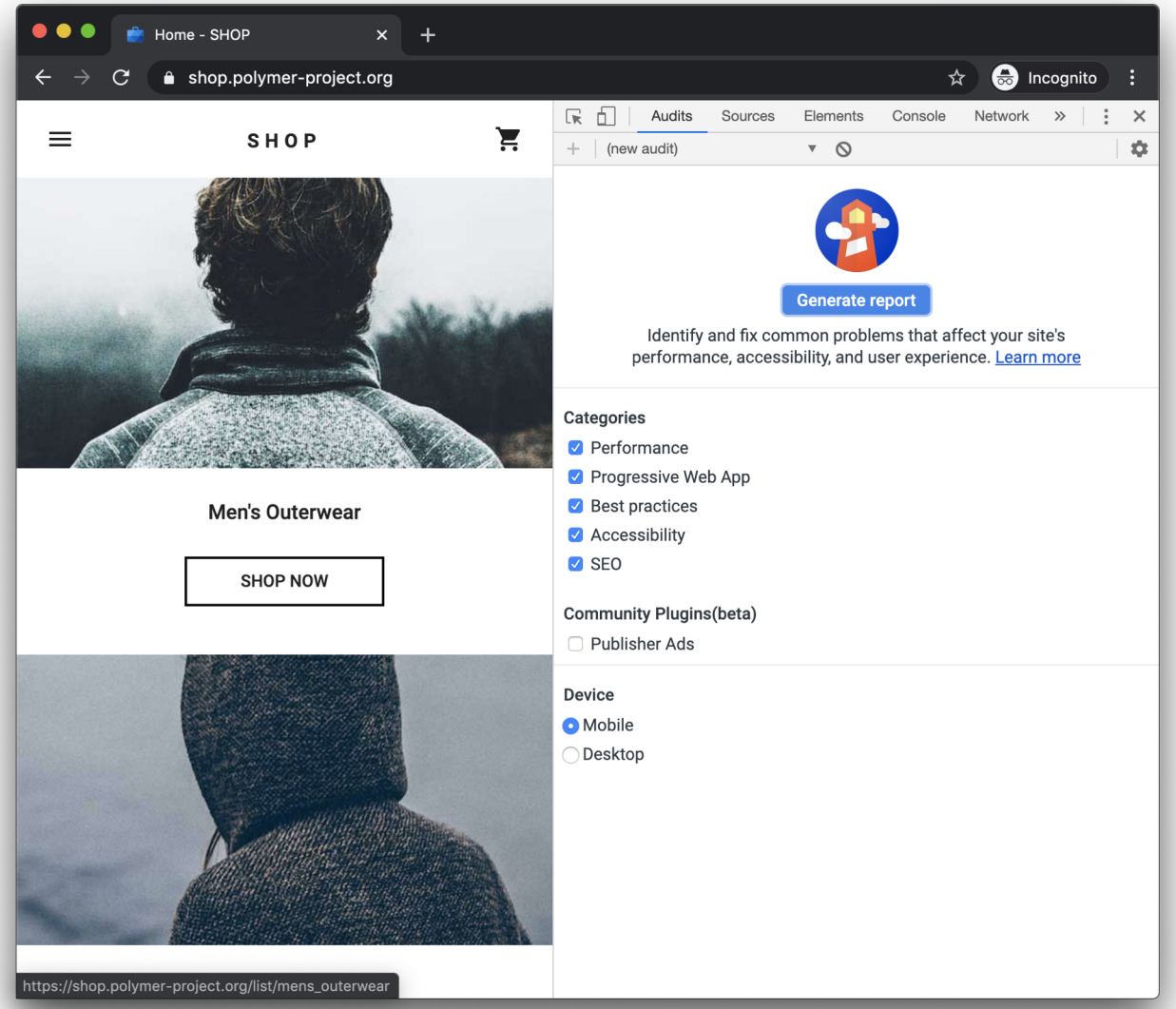
- If you're looking for ways to improve page load performance, don’t start with the Network panel.

- There are many types of load performance issues that aren't related to network activity.

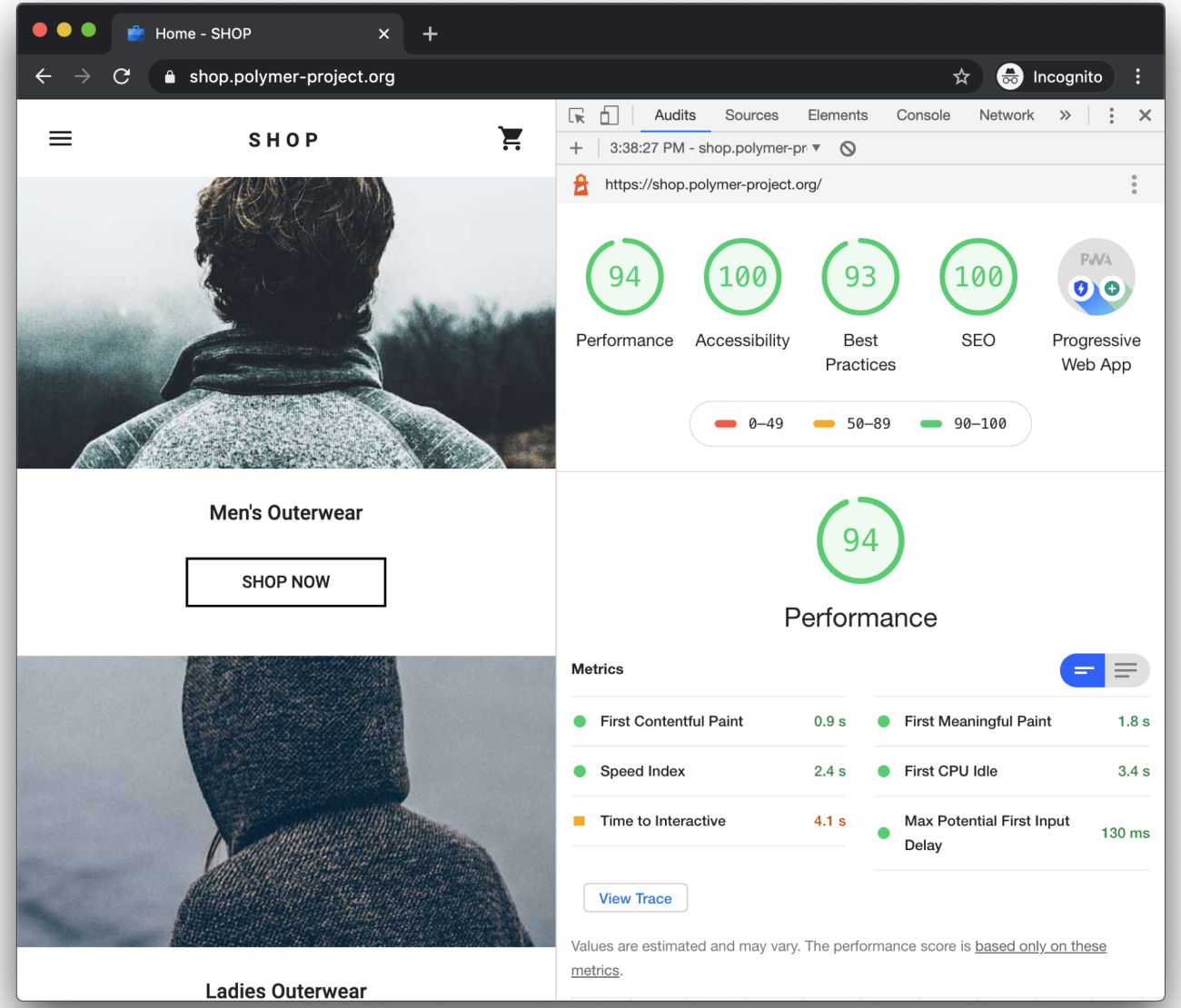
- Start with the **Audits (Lighthouse)** panel instead **because it gives you targeted suggestions on how to improve your page**.

Lighthouse

- **Lighthouse** is an open-source, automated tool for improving the quality of web pages.
- You can run it against any web page, public or requiring authentication.
- It has audits for performance, accessibility, progressive web apps, SEO and more.
- You can run Lighthouse in Chrome DevTools.

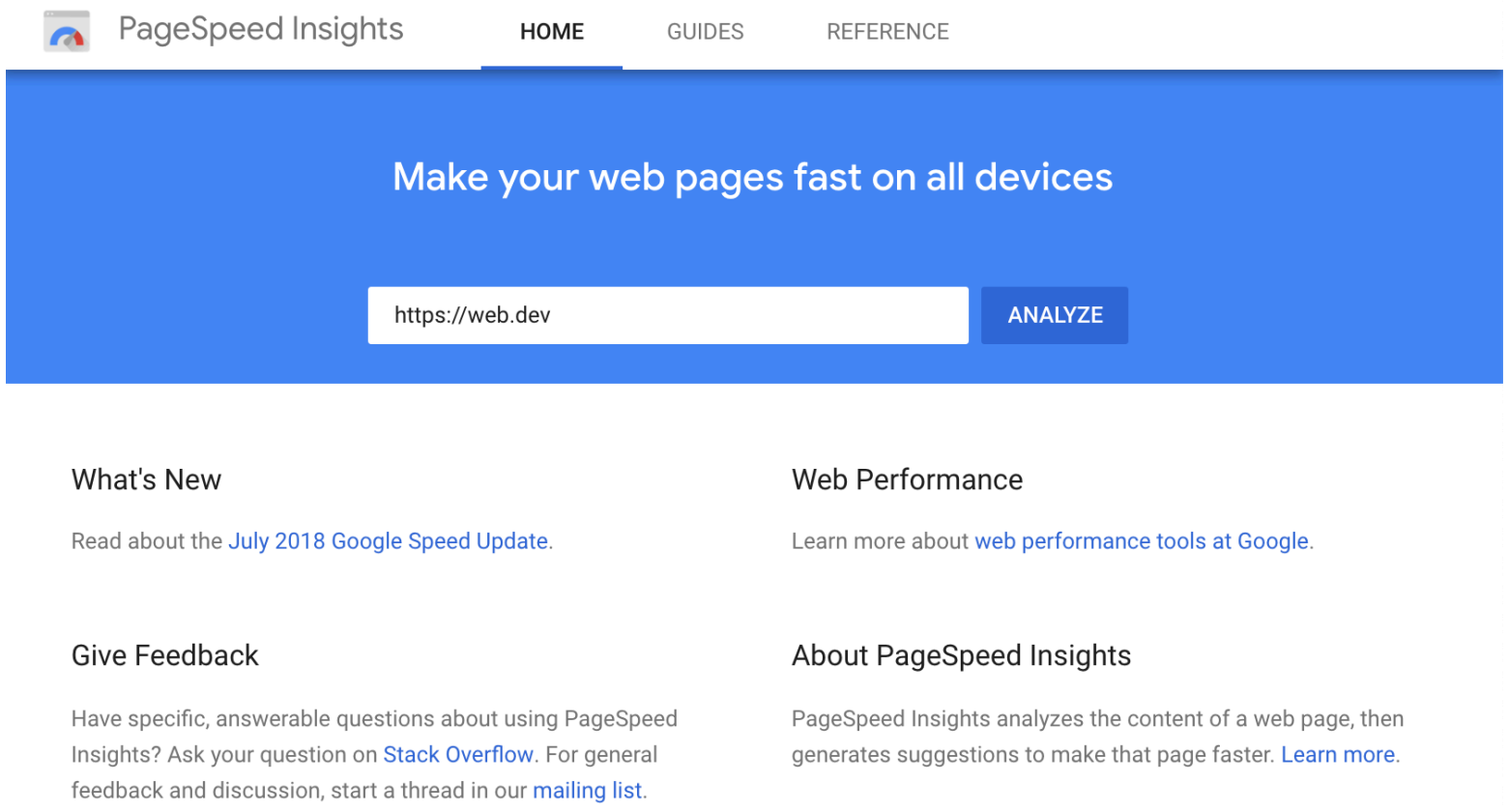


- You give Lighthouse a URL to audit, it runs a series of audits against the page, and then it generates a report on how well the page did.
- From there, use the failing audits as indicators on how to improve the page.
- Each audit has a reference doc explaining why the audit is important, as well as how to fix it.



Run Lighthouse on PageSpeed Insights

- Navigate to **PageSpeed Insights** website.
- Enter a web page URL.
- Click **Analyze**.



Assignment

Complete the **Optimize website speed** tutorial on Chrome DevTools website:

https://developer.chrome.com/docs/devtools/speed/get-started/

Contact

- Luis Beltrán
- Email: beltran_prieto@utb.cz
- WhatsApp: +420 7764 236 25

- Where can you find me:
  - Fridays U51/519 office (2 – 4 pm)

- On-line consultation (MS Teams, WhatsApp) can be done at any time.