

# What are HTTP Headers?

---

An [HTTP header](#) allows clients and servers to pass *additional* information with each request or response. Headers are just case-insensitive [key-value pairs](#) that pass additional [metadata](#) about the request or response.

HTTP requests from a web browser automatically carry with them many headers, including but not limited to:

- The type of client (e.g. Google Chrome)
- The Operating system (e.g. Windows)
- The preferred language (e.g. US English)

As developers, we can also define custom headers in each request.

## Headers API

The [Headers](#) API allows us to perform various actions on our request and response headers such as retrieving, setting, and removing them. We can access the headers object through the [Request.headers](#) and [Response.headers](#) properties.

## Using the Browser's Developer Tools

Modern web browsers offer developers a powerful set of developer tools. The Developer Tools are a front-end web developer's best friend! For example, using the dev tools you can:

- View the web page's JavaScript console output
- Inspect the page's HTML, CSS, and JavaScript code
- View network requests and responses, along with their headers.

The method for accessing dev tools varies from browser to browser.

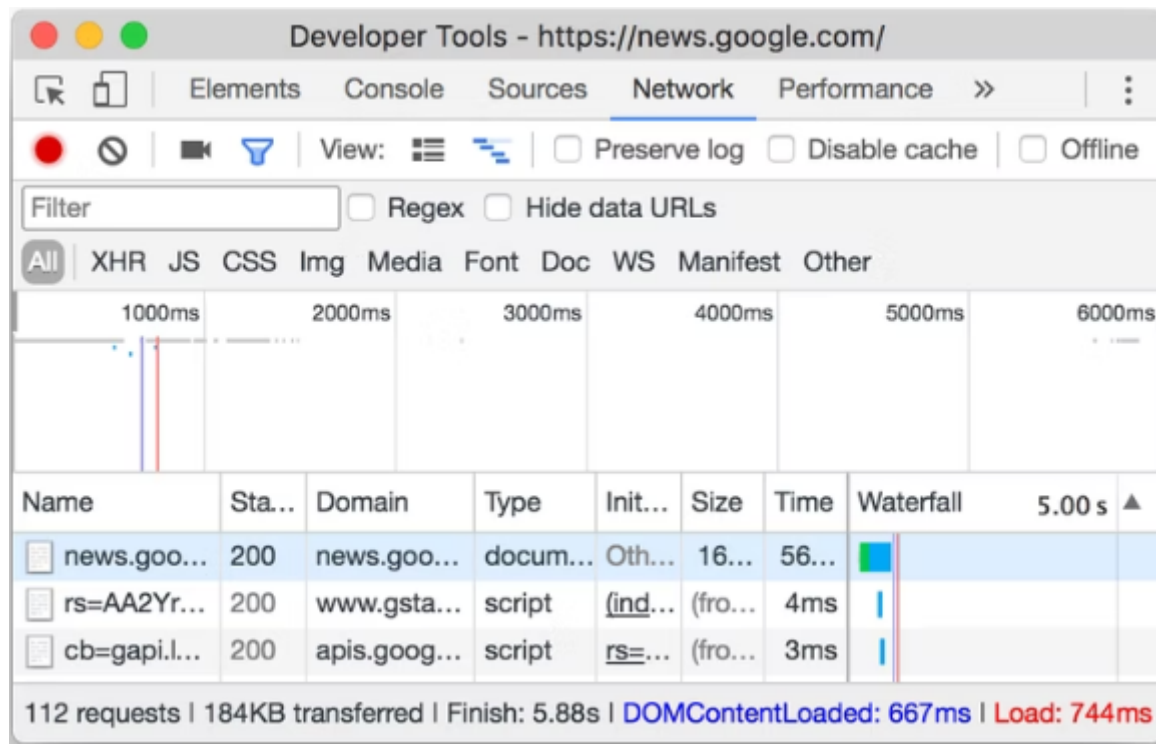
Keyboard Shortcuts:

- Windows: [ctrl + shift + I](#) or [F12](#)
- MacOS: [cmd + option + I](#)

In most browsers, you can just right-click anywhere on a web page and click the "inspect" option. Follow this [link](#) for more info on how to access dev tools.

## The network tab

While all of the tabs within the dev tools are useful, we will focus on the network tab in this chapter so we can play with HTTP headers. The network tab monitors your browser's network activity and records all of the requests and responses the browser is making, including how long each of those requests and responses takes to fully process.



## Why are headers useful?

Headers are useful for several reasons from design to security, but most often headers are used for [metadata](#) about the request or response itself. For example, let's say we wanted to ask for a player's level from the Jello server. We need to send that player's ID to the server so it knows which player to send back the information for. That ID is my request, it's not information about my request.

[Authentication](#) is a common use case for headers. If I ask Jello to complete a project, I need to provide authentication information that I'm logged in, but that auth info isn't the request itself, it's just additional information about the request.

## Network Tab Practice

1. Open your browser's Dev Tools
2. Navigate to the Network tab at the top.
3. Once you've opened the network tab, refresh this page.

Poke around through the different requests that you see. Notice that you can select a request and see its request and response headers. Request headers are sent from your browser to the server. Response headers are the headers sent back from the server to your browser. You will use the information you find within the headers tab to answer the following questions.

