

# Debugging the web

Larene Le Gassick  
Independent Software Consultant (ex-ThoughtWorks)

**email:** [contact@larene.dev](mailto:contact@larene.dev)

# Acknowledgement of country

**email: [contact@larene.dev](mailto:contact@larene.dev)**

Myth:  
Most software in the world  
is built with best-practices.

A whirlwind tour of all the tools  
that I personally use when  
building software.

I've learnt about these from  
working with many teams.

# Outline

- Oh no, our app, it's broken!
- Why are my changes not showing up?
- Exploring and testing APIs
- Why do I keep getting CORS errors?
- Why is my website not online?

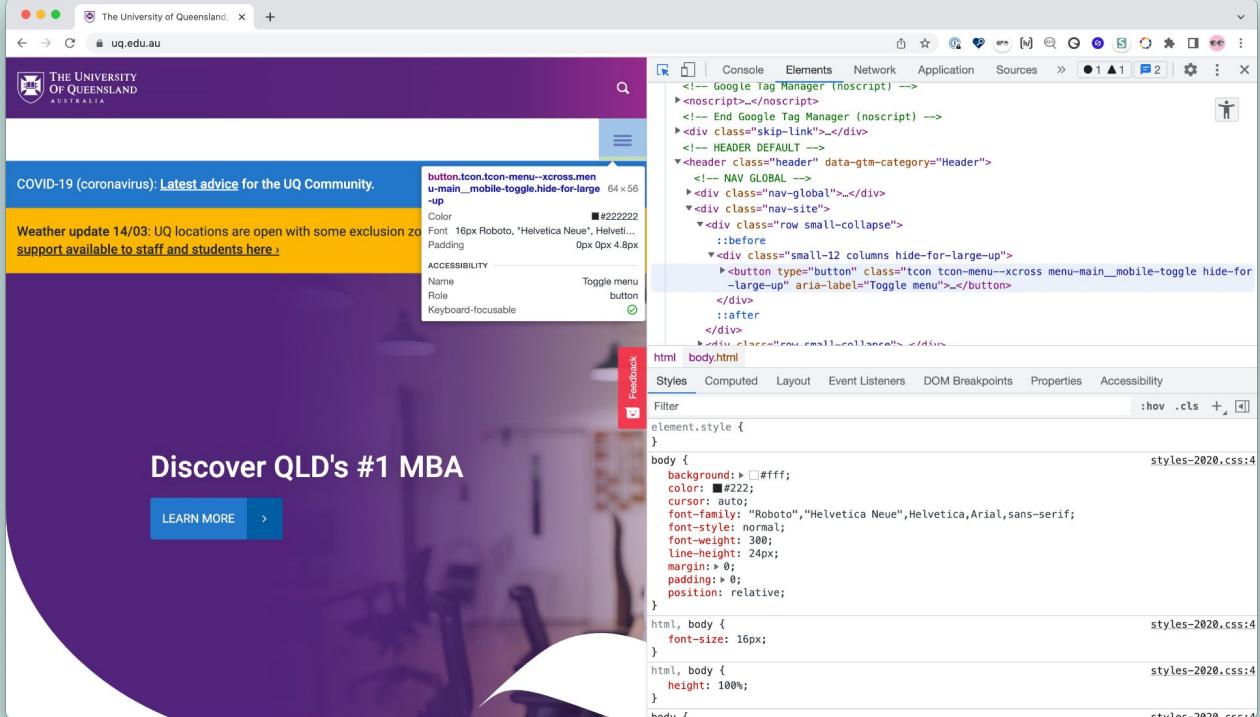
# Oh no, our app, it's broken!

email: [contact@larene.dev](mailto:contact@larene.dev)

# Browser Developer Tools

- All browsers have **built-in developer tools**.
- Many of you will have used them before.
- Worth learning how to use them properly!
- Not just for front-end developers
  - API Developers use the Network Inspector frequently

# Inspect Elements



email: [contact@larene.dev](mailto:contact@larene.dev)

# How to open Inspect Elements tool?

Using Google Chrome, but very similar in other browsers.

- Right-click > Inspect.
- Command + Option + i for Mac, Ctrl + Shift + i for Windows.
- Firefox: "Inspector" Tab.

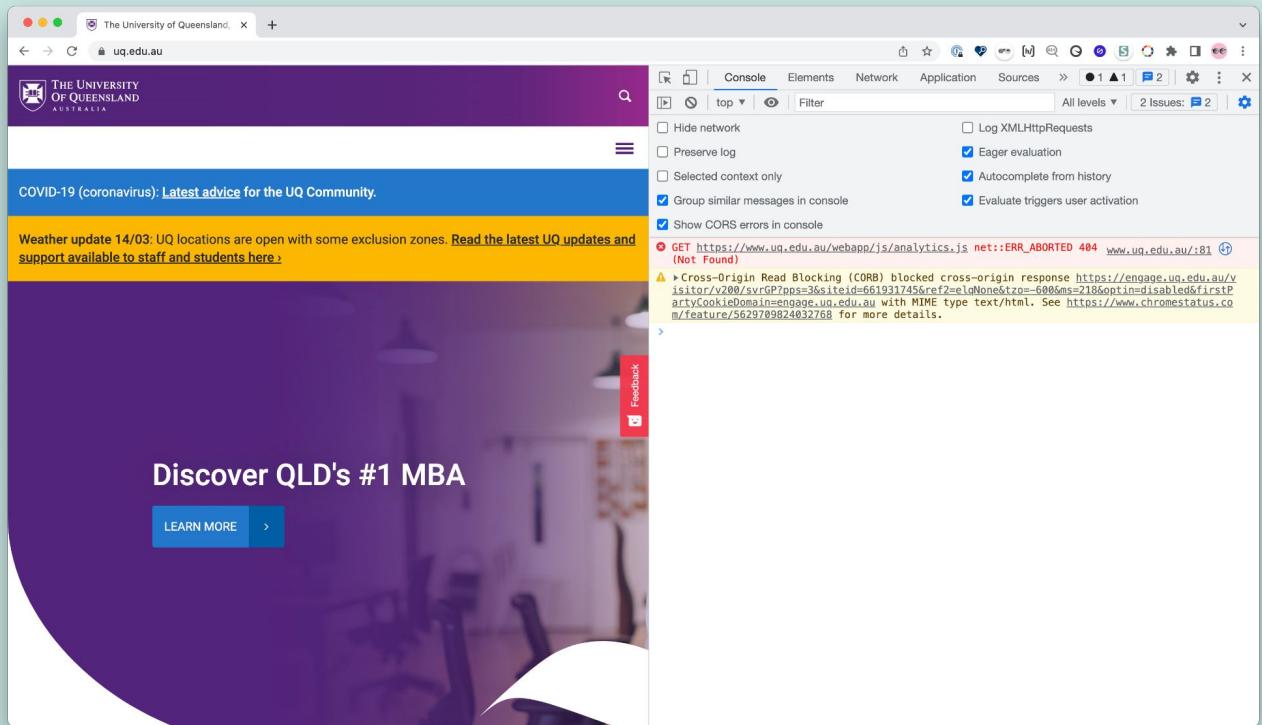
When Elements tab is open, you can:

-  Select an element on the page (Command + Shift + C)
-  Toggle the device/response view (Command + Shift + M)

# Use Inspect Elements when...

- Debugging or inspecting CSS.
- Seeing what JS is doing to the DOM when you interact with the page.
- Quickly test CSS or HTML changes.
- Test the app by simulating another device, for example, an iPhone.
- Different to "View source code".
  - Easter eggs! Try View Source on <https://theoatmeal.com> (warning: swear words!)

# Developer Console



email: [contact@larene.dev](mailto:contact@larene.dev)

# How to open the Developer Console?

Using Google Chrome, but very similar in other browsers.

- Right-click > Inspect, select "Console" tab.
- Command + Option + j for Mac, Ctrl + Shift + j for Windows.

Make sure to refresh the page while it's open, to get all the logs.

If you want to keep the logs between page changes/refreshes, check the Settings > "Preserve log" checkbox.

# Use the Developer Console to...

- Look for JS warnings and errors (that maybe stopping execution).
- Inspect global JS variables.
- Use it to write JS functions to extract information.
- Use it to quickly test JS.
  - also use a REPL like CodeSandbox for this.
- More easter eggs!
  - Search for "text adventure" into Google search, and open developer console on the results page.

# Network

The screenshot shows a web browser window for "The University of Queensland" at "uq.edu.au". The page content includes a purple header with the university logo, a blue banner about COVID-19 advice, a yellow banner about weather updates, and a purple background image of a classroom. A "Discover QLD's #1 MBA" call-to-action is visible. The Network tab of the developer tools is active, displaying a timeline and a detailed list of network requests. The timeline shows several requests starting around 1000 ms and continuing. The detailed list shows 69 requests, mostly from "www.go...". Initiators include "analytics.js:44", "hotjar-2849...", and various script and font files. The "Waterfall" column shows the duration of each request.

Path	Status	Domain	Type	Initiator	Size	T.	Waterfall
/j/collect	200	www.go...	xhr	analytics.js:44	22 B	1...	
/modules.7d3f95230...	200	script.h...	script	hotjar-2849...	(disk ...	4...	
/box-acca23410e696...	200	vars.hotj...	document	(disk ...	2...		
/j/collect	200	stats.g...	xhr	analytics.js:44	29 B	2...	
/j/collect	200	stats.g...	xhr	analytics.js:44	29 B	2...	
/preact-incoming-fee...	200	script.h...	script	modules.7d...	(disk ...	3...	
/font-hotjar_6.65042...	200	script.h...	script	(index)	(me...	0...	
/s/0.6.4/clarity.js	200	e.clarity...	script	1370000632...	(disk ...	3...	
/tr/	200	www.fa...	document	(index)	15 B	2...	
· /ads/ga-audiences	200	www.go...	gif	analytics.js:31	64 B	1...	
· /ads/ga-audiences	200	www.go...	gif	analytics.js:31	63 B	2...	
· /ads/ga-audiences	200	www.go...	gif	analytics.js:31	64 B	1...	
· /ads/ga-audiences	200	www.go...	gif	analytics.js:31	63 B	2...	
/collect	204	e.clarity...	xhr	clarity.js:2	174 B	8...	
/c.gif	302	c.clarity...	/ Red...	1370000632...	322 B	4...	
/favicon.ico	200	www.uq...	x-icon	Other	1.2 kB	1...	
/favicon.ico	200	www.uq...	x-icon	Other	1.2 kB	9...	
/c.gif	302	c.bing.c...	/ Red...	c.gif	405 B	1...	
· /c.gif	200	c.clarity...	gif	c.gif	281 B	1...	
/collect	204	e.clarity...	xhr	clarity.js:2	48 B	2...	
/collect	204	e.clarity...	xhr	clarity.js:2	48 B	2...	

69 requests | 1.5 MB transferred | 2.8 MB resources | Finish: 6.62 s | DOMContentLoaded: 450 ms | Load: 1.81

email: contact@larene.dev

# How to open the Network tab?

Using Google Chrome, but very similar in other browsers.

- Right-click > Inspect, then select "Network" tab.
- `Command + Option + i` for Mac, `Ctrl + Shift + i` for Windows, then select "Network" tab.

Make sure to refresh the page while it's open, to get all the network calls.

If you want to keep the logs between page changes/refreshes, check the "Preserve log" checkbox.

# Use the Network tab to...

- Understand what requests the current app is making.
  - And the responses to those request.
- Troubleshoot API requests.
- Disable browser cache.
  - More on cache, later.
- Throttle the page load speed.
- Copy the request as a script or code.
  - For example, right-click Request > Copy as cURL.
  - We'll see how to use this in Postman later.

# Application

The University of Queensland

COVID-19 (coronavirus): Latest advice for the UQ Community.

Weather update 14/03: UQ locations are open with some exclusion zones. [Read the latest UQ updates and support available to staff and students here](#).

Discover QLD's #1 MBA

LEARN MORE >

Feedback

Application

Name	Value	D	P	E	S	H	S	S	P	P
CJ5vpP9M4IVOPYB...	.... /	2...	5...	✓	✓	✓	N..	H..		
J0r-GoVi_bvCnLJ...	.... /	2...	4...						H..	
HSID	A8252m647fjxytb...	.... /	2...	2...	✓					H..
_S...	IAhh-7MYUNaevIX...	.... /	2...	8...	✓	✓	N..	H..		
Storage										
Local Storage										
Session Storage										
IndexedDB										
Web SQL										
Cookies										
https://www.uq.edu.au										
vars.hotjar.com										
Trust Tokens										
Cache										
Cache Storage										
Back/forward cache										
Background Services										
Background Fetch										
Background Sync										
Notifications										
Payment Handler										
Periodic Background Sync										
Push Messaging										
Reporting API										
Frames										
top										

Select a cookie to preview its value

email: [contact@larene.dev](mailto:contact@larene.dev)

# How to open the Application tab?

Using Google Chrome, but very similar in other browsers.

- Right-click > Inspect, then select "Network" tab.
- Command + Option + i for Mac, Ctrl + Shift + i for Windows, then select "Application" tab.
- Firefox: "Storage" tab.

Only really need:

- Cookies
- Local Storage
- Session Storage

# Use the Application tab when...

- Inspecting, deleting, or changing
  - cookies,
  - session storage key-values,
  - local storage key-values.
- Troubleshooting service workers & browser database (IndexedDB)
  - typically only for progressive web apps (PWAs).

# Sources

The screenshot shows a web browser window displaying the University of Queensland's website ([uq.edu.au](http://uq.edu.au)). The page features a purple header with the university's logo and a navigation menu. A blue banner at the top provides COVID-19 information. A yellow banner below it contains a weather update. The main content area has a purple background and features a large image of a classroom. A call-to-action button says "Discover QLD's #1 MBA" with a "LEARN MORE" link. The browser's developer tools are open, specifically the "Sources" tab, which shows the raw HTML and JavaScript code for the page. The code includes meta tags for search engines, CSS styles, and various JavaScript files like bower\_components, fontawesome/fonts, jquery, and uq-standard-global-menu.js.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
<!--[if lt IE 7]><html class="lt-ie9 lt-ie8 lt-ie7"><![endif]-->
<!--[if IE 7]><html class="lt-ie9 lt-ie8"><![endif]-->
<!--[if IE 8]><html class="lt-ie9"><![endif]-->
<!--[if gt Ie 8]>!-->
<html>
<head>
<meta content="IE=Edge" http-equiv="X-UA-Compatible"/>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<meta name="ROBOTS" content="ALL"/>
<meta name="title" content="The University of Queensland, Australia"/>
<meta name="description" property="og:description"/>
<meta name="keywords" content="University of Queensland, Australia"/>
<meta name="language" content="en" />
<meta name="msvalidate.01" content="20077888"/>
<meta name="msvalidate.01" content="D347AB6"/>
<meta property="og:title" content="The University of Queensland, Australia"/>
<meta property="og:description" content="The University of Queensland, Australia"/>
<meta property="og:image" content="https://www.uq.edu.au/images/the-university-of-queensland-australia/logo"/>
<meta property="og:page" content="https://www.uq.edu.au/v9/index.html"/>
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "EducationalOrganization",
  "name": "The University of Queensland",
  "alternateName": "UQ",
  "url": "https://www.uq.edu.au",
  "logo": "https://static.uq.net.au/v9/images/uq/logo.png"
}

```

email: [contact@larene.dev](mailto:contact@larene.dev)

# How to open the Sources tab?

Using Google Chrome, but very similar in other browsers.

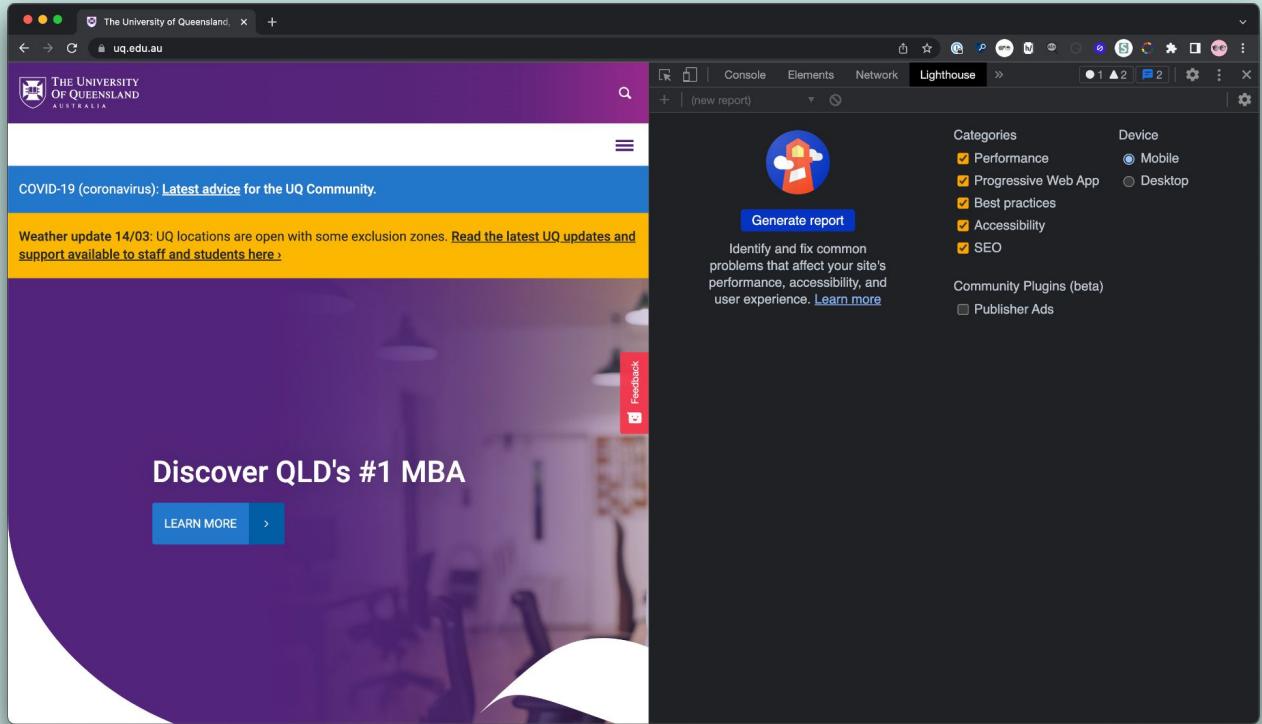
- Right-click > Inspect, then select "Network" tab.
- `Command + Option + i` for Mac, `Ctrl + Shift + i` for Windows, then select "Sources" tab.
- This is called the "Debugger" tab in Firefox.

# Use the Sources tab to...

- debug JavaScript in a live site.
- inspect website file/folder structure.

You usually get to the Sources tab by clicking on logs in the Console.

# Lighthouse



email: contact@larene.dev

# How to open the Lighthouse tab?

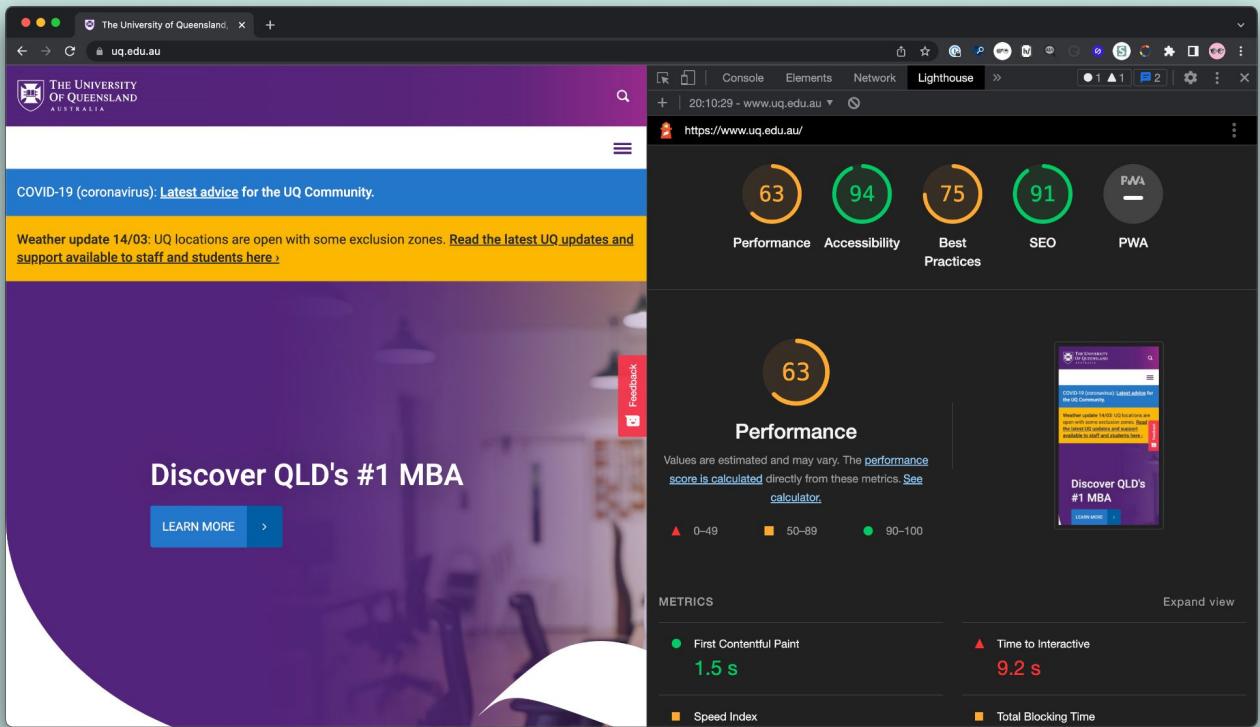
Lighthouse is **only in Google Chrome**.

- Right-click > Inspect, then select "Lighthouse" tab.
- Command + Option + i for Mac, Ctrl + Shift + i for Windows, then select "Lighthouse" tab.

# Why use Lighthouse tab?

- A great way to be introduced to web best practices in
  - accessibility,
  - SEO,
  - performance,
  - other best practices.
- Run these often on your own websites, and websites you like!
- A note on accessibility
  - a 100 Lighthouse score doesn't mean it's accessible, automated tests can only detect 30-50% of accessibility issues. See my past lectures in Accessibility ;).

# UQ Lighthouse Score



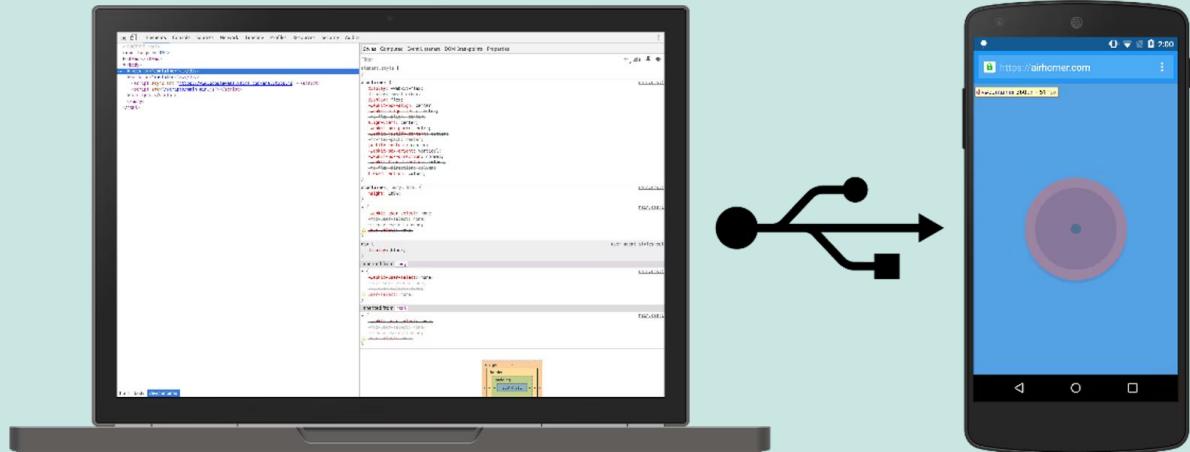
email: [contact@larene.dev](mailto:contact@larene.dev)

# Other tabs

- Performance, Memory, Security tabs:
  - Rarely used it, but good to know it's there.
  - Use to debug really slow complex websites. What is loading when? Is it parallel or serial?
- Advanced UI debugging
  - For example, More tools > Rendering > Paint flashing.
- Install other browser extensions
  - Framework-specific tools. React Developer Tools if you're doing React, for example.
  - Accessibility testing tools.

# Debugging your mobile browser

- Debugging iOS Safari on a Mac
- Debugging Android Chrome



email: [contact@larene.dev](mailto:contact@larene.dev)

# Why are my changes not showing up?

email: [contact@larene.dev](mailto:contact@larene.dev)

# Why are my changes not showing up?

- Did you save the file?
- Did you refresh the page?
- Are you hitting the right URL?
- Are you sure you saved the file??
- It's probably **cache!**

# What is cache?

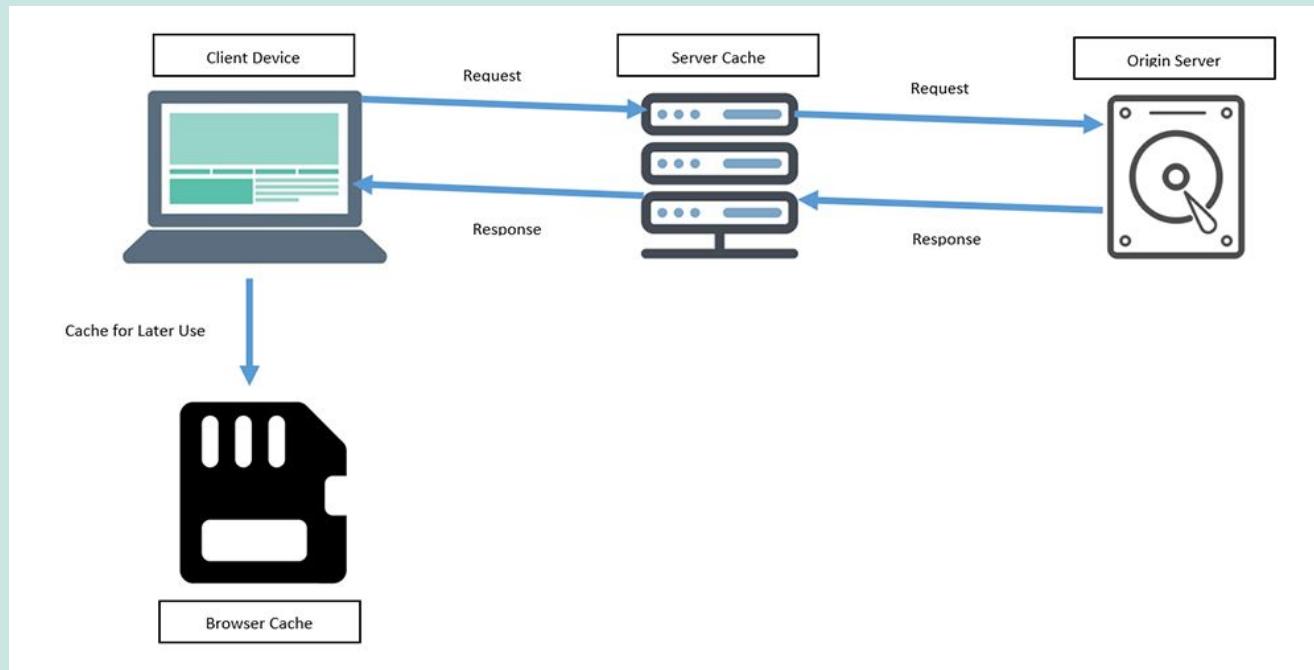
In web development, cache is found in a few places:

- Your browser
- Proxy-server/CDN
- Application Server or Middleware
- Your server-side code

Important cache property:

- **Expiry** — "When should we go fetch a fresh copy?"

# Where can caching happen?



Source: [Understanding Caching and How It Impacts Site Speed - Ezoic](#)

email: [contact@larene.dev](mailto:contact@larene.dev)

# How to know it's cache?

- Browser cache
  - stored in temporary files or memory on your computer
  - network tab will show "cached" in the request information
- Proxy-server/CDN
  - stored in memory on the server
  - sometimes shows in the response headers, for example CloudFront

Don't forget to check in your application!

- Middleware or Application Server (Tomcat in Java, for example)
- Your server-side code (if you use a Cache library, for example)

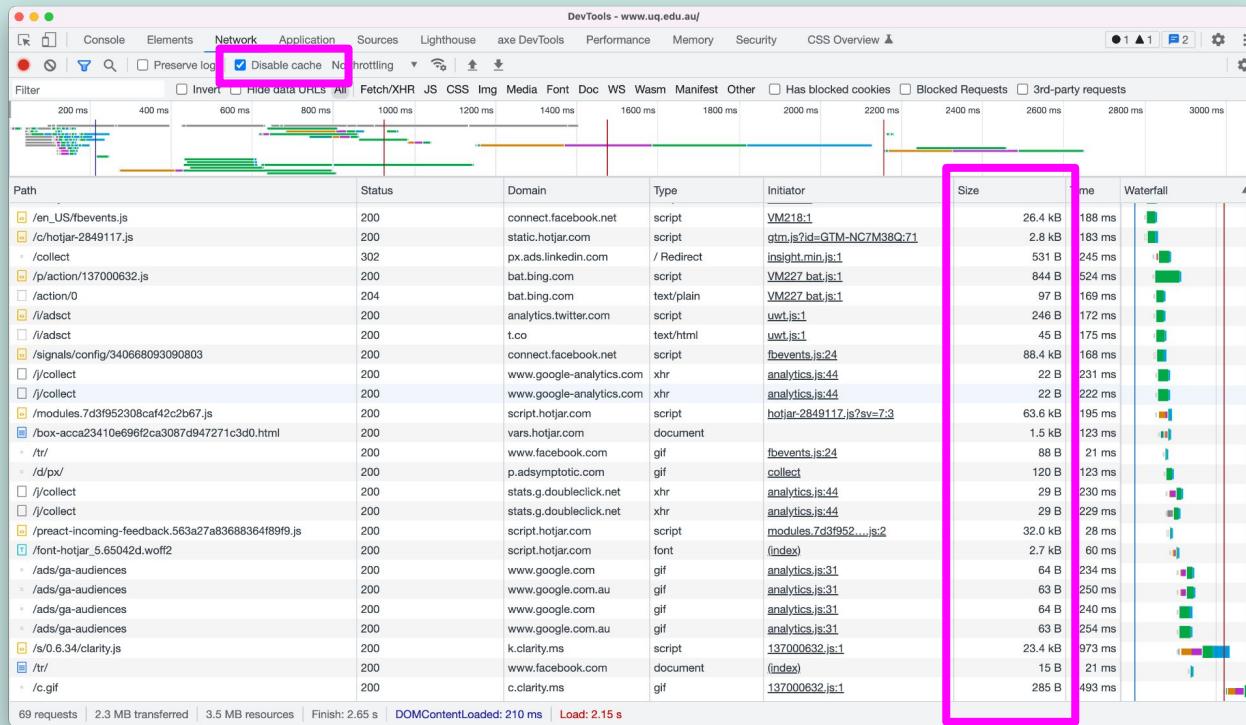
# With browser cache enabled

The screenshot shows the Network tab in the Chrome DevTools interface. At the top left, there is a checkbox labeled "Preserve log". Below the header, there is a table with columns: Path, Status, Domain, Type, Initiator, Size, Time, and Waterfall. The "Size" column is highlighted with a pink box. The table lists numerous network requests, including scripts from connect.facebook.net, images from bat.bing.com, and various CSS and JS files from www.uq.edu.au. The "Size" column shows file sizes such as 793 B, 122 B, and 170 ms. The "Time" column shows response times like 459 ms, 168 ms, and 22 ms.

Path	Status	Domain	Type	Initiator	Size	Time	Waterfall
/en_US/fbevents.js	200	connect.facebook.net	script	VM130:1	(memory cache)	0 ms	
/p/action/137000632.js	200	bat.bing.com	script	VM123 bat.js:1	793 B	459 ms	
/action/0	204	bat.bing.com	text/plain	VM123 bat.js:1	122 B	168 ms	
/modules.7d3f952308caf42c2b67.js	200	script.hotjar.com	script	hotjar-2849117.js?sv=7:3	(memory cache)	0 ms	
/signals/config/340668093090803	200	connect.facebook.net	script	fbevents.js:24	(memory cache)	0 ms	
/actionp/0	204	bat.bing.com	ping	VM123 bat.js:1	122 B	170 ms	
/tr/	200	www.facebook.com	gif	fbevents.js:24	88 B	22 ms	
/visitor/v200/svrGP	200	engage.uq.edu.au	text/html	elqCfg.min.js:1	0 B	298 ms	
/highlighter.js	200	lhdroppojmgadmnindn...	script	content.bundle.js:1	17.9 kB	7 ms	
/iadst	200	analytics.twitter.com	script	uwt.js:1	245 B	173 ms	
/box-acca23410e696f2ca3087d947271c3d0.html	200	vars.hotjar.com	document		(disk cache)	4 ms	
/iadst	200	t.co	text/html	uwt.js:1	50 B	173 ms	
/dp/x	200	p.adsymptotic.com	gif	collect	120 B	125 ms	
/preact-incoming-feedback.563a27a83688364fb9f9.js	200	script.hotjar.com	script	modules.7d3f952...js:2	(memory cache)	0 ms	
/font-hotjar.5.65042d.woff2	200	script.hotjar.com	font	(index)	(memory cache)	0 ms	
/s/0.6.33/clarity.js	200	h.clarity.ms	script	137000632.js:1	(disk cache)	2 ms	
/c.gif	200	c.clarity.ms	gif	137000632.js:1	104 B	170 ms	
/favicon.ico	200	www.uq.edu.au	x-icon	Other	1.2 kB	10 ms	
/favicon.ico	200	www.uq.edu.au	x-icon	Other	1.2 kB	8 ms	
/collect	204	h.clarity.ms	xhr	clarity.js:2	48 B	224 ms	
/tr/	200	www.facebook.com	document	Other	15 B	21 ms	
/collect	204	h.clarity.ms	xhr	clarity.js:2	48 B	270 ms	
/v11/logos/corporate/uq-lockup-landscape--reversed.svg	200	static.uq.net.au	svg+xml	styles-2020.css	(disk cache)	2 ms	
/images/vc-home-banner.jpg	200	www.uq.edu.au	jpeg	styles-2020.css	508 kB	49 ms	
/collect	204	h.clarity.ms	xhr	clarity.js:2	48 B	252 ms	

email: contact@larene.dev

# With browser cache disabled



email: [contact@larene.dev](mailto:contact@larene.dev)

# Proxy-server cache, example using CloudFront

The screenshot shows the Network tab in Google Chrome DevTools for the URL [www.rolexsydneyhobart.com/](https://www.rolexsydneyhobart.com/). The timeline shows multiple requests being made. A specific request for `/css/bwps.min.css` is selected.

**Request Headers:**

- X-Cache: Hit from cloudfront
- Accept: text/css,\*/\*;q=0.1
- Accept-Encoding: gzip, deflate, br
- Accept-Language: en-AU,en-US;q=0.9,en;q=0.8
- Cache-Control: no-cache

**Response Headers:**

- Accept-Ranges: bytes
- Age: 68164
- Connection: keep-alive
- Content-Length: 555238
- Content-Type: text/css
- Date: Sat, 26 Mar 2022 12:13:15 GMT
- ETag: "1d823ebd1bcc766"
- Last-Modified: Thu, 17 Feb 2022 10:47:55 GMT
- Server: Kestrel
- Via: 1.1 00f0469d54a973389150a36c64065326.cloudfront.net (CloudFront)
- X-Amz-Cf-Id: VVDIC5ZReAbmrxr4qkBoJbE41D011Q0n0HuHKN1T-2vu5bhfh4e9-rQ==
- X-Amz-Cf-Delivery-Point: P1

**Timing Panel:**

- Request: 100 ms
- Latency: 200 ms
- Connect: 300 ms
- Processing: 400 ms
- Content: 500 ms
- SSL: 600 ms
- Round Trip: 700 ms
- Latency: 800 ms
- Content: 900 ms
- SSL: 1000 ms
- Round Trip: 1100 ms
- Latency: 1200 ms
- Content: 1300 ms
- SSL: 1400 ms
- Round Trip: 1500 ms
- Latency: 1600 ms
- Content: 1700 ms
- SSL: 1800 ms
- Round Trip: 1900 ms
- Latency: 2000 ms

**Path:**

- /
- /css/bwps.min.css
- /ajax/libs/jquery/1.7.0/jquery.min.js
- /js/rshyr.min.js
- /lib/body-scroll-lock/bodyScrollLock.js
- /css/rshyr/media/mediaelement-and-play..
- /css/rshyr/media/mediaelementplayer.mi..
- /css/jquery.fancybox.min.css
- /rolex/css/rshyr-rolex.css
- /mapbox-gl-js/v0.41.0/mapbox-gl.css
- /mapbox-gl-js/v0.41.0/mapbox-gl.css
- /js/yachttracker\_home\_mb.min.js
- /rolex/js/modernizr.custom.min.js
- /rolex/js/jquery.countdown.js
- /rolex/js/rolex-core-1.4.2.js
- /rolex/js/timezones.js
- /rolex/js/listofdates.js
- /rolex/js/watch.js
- /axe-versions/lastest/axe.js
- /highlighter.js
- /gtag.js
- /crops(164,105)/media/3443471/syho21a..
- /crops(335,214)/media/3443471/syho21a..
- /crops(164,105)/media/3443630/jules-an..

122 requests 6.3 MB transferred 6.9 MB

email: [contact@larene.dev](mailto:contact@larene.dev)

# Exploring and testing APIs

email: [contact@larene.dev](mailto:contact@larene.dev)

# About Postman

- Originally a tool for testing APIs so that you could stop using 'cURL' commands. Now has many more features.
- Used heavily in industry.
- Great for
  - working out how to use an API before you code it into your client-side or server-side app,
  - testing and documenting your API,
  - troubleshooting CORS.



email: [contact@larene.dev](mailto:contact@larene.dev)

# API 1: SWAPI API (no auth)

[swapi.dev](https://swapi.dev)

email: [contact@larene.dev](mailto:contact@larene.dev)

# SWAPI

The Star Wars API

(what happened to swapi.co?)

All the Star Wars data you've ever wanted:

**Planets, Spaceships, Vehicles, People, Films and Species**

From all **SEVEN** Star Wars films

**Now with The Force Awakens data!**

## Try it now!

<https://swapi.dev/api/>

people/1/

request

Need a hint? try *people/1/* or *planets/3/* or *starships/9/*

API 2: Spotify API (has auth)  
[developer.spotify.com](https://developer.spotify.com)

email: [contact@larene.dev](mailto:contact@larene.dev)

Step 0 - go to [Spotify Web API](#) docs

# Platform Documentation

Here you'll find handy documentation about our APIs and SDKs.

Choose below, or scroll down to select by feature.



WEB API



WEB  
PLAYBACK SDK



IOS



ANDROID



WIDGETS



COMMERCIAL  
HARDWARE

# Step 1 - copy the URL into Postman.

The screenshot shows the Postman application interface. At the top, the URL `https://api.spotify.com/v1/albums/id` is entered into the address bar. Below it, a GET request is selected, and the URL `https://api.spotify.com/v1/albums/3jk6oyfKCny48ZYb1CPjLa` is displayed in the request field. A large blue "Send" button is visible. The "Params" tab is active, showing a single entry under "Query Params": "Key" and "Value". In the main body area, the response status is shown as 400 Bad Request with a message: "Only valid bearer authentication supported".

https://api.spotify.com/v1/albums/id

GET https://api.spotify.com/v1/albums/3jk6oyfKCny48ZYb1CPjLa

Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body 400 Bad Request 322 ms 879 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "error": {
3      "status": 400,
4      "message": "Only valid bearer authentication supported"
5    }
6 }
```

E-mail: contact@larene.dev

# Step 2 - create a Spotify account, and create an app.

The screenshot shows the Spotify for Developers Dashboard. At the top, there is a navigation bar with links: Spotify for Developers, DISCOVER, DOCS, CONSOLE, COMMUNITY, DASHBOARD (which is underlined in green), and USE CASES. Below the navigation bar, the word "Dashboard" is prominently displayed in large, bold, black font. On the right side of the dashboard, there are two buttons: "CREATE AN APP" (in a green rounded rectangle) and "LOGOUT" (in a white rounded rectangle). The main area of the dashboard displays three app cards, each with a title, a CLIENT ID, and a brief description.

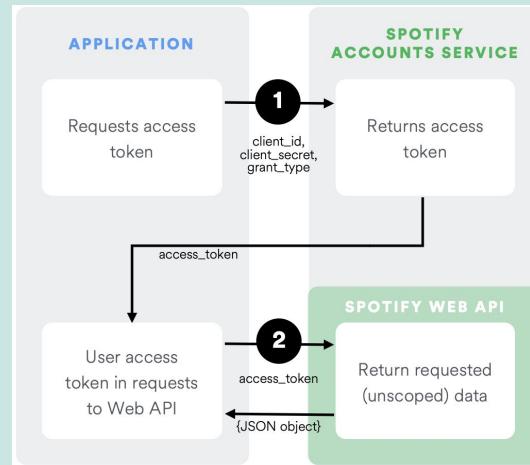
- XMas Party**  
CLIENT ID 3d415a521f734f6c9345904767e7f98b  
Testing sorting and scheduling stuff
- CSSE6400 Demo**  
CLIENT ID c84576f5b0e84f159525ecafcc3a6336e  
Using Postman to explore Spotify API
- My New App**  
>\_

ene.dev

# Step 3 - Find the docs on how to authenticate.

## Client Credentials Flow

The Client Credentials flow is used in server-to-server authentication. Since this flow does not include authorization, only endpoints that do not access user information can be accessed.



**email:** contact@larene.dev

# Step 3, continued.

## Request authorization

The first step is to send a `POST` request to the `/api/token` endpoint of the *Spotify OAuth 2.0 Service* with the following parameters encoded in `application/x-www-form-urlencoded`:

REQUEST BODY PARAMETER	VALUE
grant_type	<i>Required</i> Set it to <code>client_credentials</code> .

The headers of the request must contain the following parameters:

HEADER PARAMETER	VALUE
Authorization	<i>Required</i> Base 64 encoded string that contains the client ID and client secret key. The field must have the format: Authorization: <code>Basic &lt;base64 encoded client_id:client_secret&gt;</code>
Content-Type	<i>Required</i> Set to <code>application/x-www-form-urlencoded</code> .

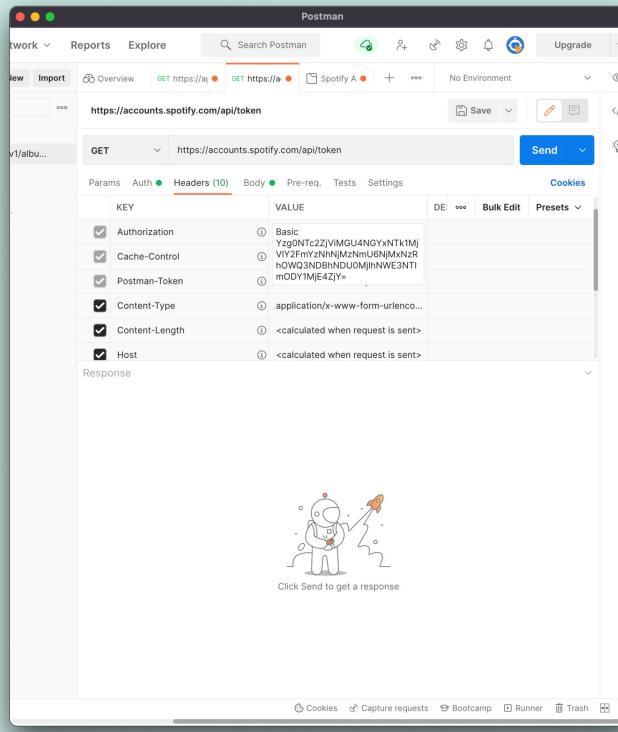
# Step 4 - setup the auth endpoint.

The screenshot shows the Postman application interface. A GET request is being prepared to <https://accounts.spotify.com/api/token>. In the 'Body' tab, under 'x-www-form-urlencoded', there is one parameter: 'grant\_type' with the value 'client\_credentials'. The 'Send' button is visible at the top right of the request panel.

The screenshot shows the same Postman interface after configuration. The 'Auth' tab is selected, and the 'Basic Auth' type is chosen. The 'Username' field contains the value 'c84576f5b0e84ff159525ecacf3a63...' and the 'Password' field contains the value '63174a9d740a45429a5a759f8652...'. A checkbox for 'Show Password' is checked. The 'Send' button is visible at the top right.

email: contact@larene.dev

# Step 5 - double-check the headers.



The screenshot shows the Postman application interface. A GET request is being prepared to the endpoint `https://accounts.spotify.com/api/token`. The 'Headers' tab is active, displaying the following configuration:

KEY	VALUE
Authorization	Basic Yzg0NTc2ZjVlMGU4NGYxNTk1MjVlY2FmYzNhNjMzMnU6NmNzRmhOWQ3NDNBNUDl0jhjNWE3NTImODYmJ4ZjY=
Cache-Control	no-store
Postman-Token	505f3a2d-4a2e-4333-8a2a-1a0a2a2a2a2a
Content-Type	application/x-www-form-urlencoded
Content-Length	<calculated when request is sent>
Host	<calculated when request is sent>

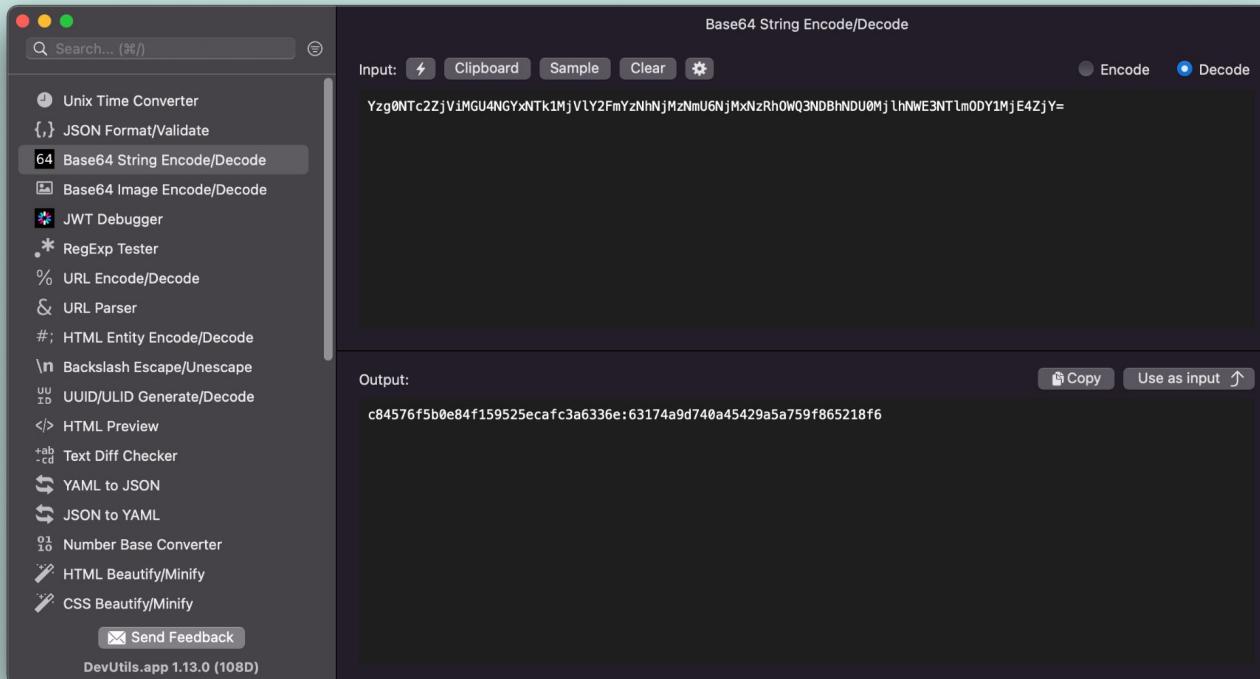
The 'Body' tab is also visible, showing a JSON payload:

```
{ "grant_type": "client_credentials", "client_id": "505f3a2d-4a2e-4333-8a2a-1a0a2a2a2a2a", "client_secret": "505f3a2d-4a2e-4333-8a2a-1a0a2a2a2a2a" }
```

The 'Send' button is highlighted in blue at the top right of the request panel. Below the request panel, there is a response placeholder area featuring a cartoon character holding a rocket.

email: [contact@larene.dev](mailto:contact@larene.dev)

# Step 5a - Base64 decoder (lots of free ones online!)



email: contact@larene.dev

# Step 6 - get the access token, and read docs on how to use it.

The screenshot shows the Postman application interface. A POST request is being made to <https://accounts.spotify.com/api/token>. The 'Auth' tab is selected, showing 'Basic Auth' parameters: Username (c84576f5b0e84f159525ecacf3a63...) and Password (63174a9d740a45429a5a759f8652...). The response status is 200 OK, with a duration of 373 ms and a size of 765 B. The response body is displayed in JSON format:

```
1 "access_token":  
2 "80810xEPbf62pk1ps6TzbVna3FwNq75c71yeui3-F-w9EtDKgztzQOrqrRe8U8QwMpd05a1A2  
3 1lypjn",  
4 "token_type": "Bearer",  
5 "expires_in": 3600
```

## How to use the Access Token

The access token allows you to make requests to the [Spotify Web API](#). To do so, you need to include the following header in your API calls:

HEADER PARAMETER	VALUE
Authorization	<i>Required</i> Valid access token following the format: <code>Bearer &lt;Access Token&gt;</code>

The following example uses `cURL` to retrieve information about a track using the [Get a track](#) endpoint:

```
curl --request GET \  
'https://api.spotify.com/v1/tracks/2TpXZ7JUBn3uw46aR7qd6V' \  
--header "Authorization: Bearer NgCXRK...MzYjw"
```

email: [contact@larene.dev](mailto:contact@larene.dev)

# Step 7 - put the access token into the Auth tab of the request.

The screenshot shows the Postman interface with a GET request to `https://api.spotify.com/v1/albums/{id}`. The **Auth** tab is selected, showing a `Bearer Token` field with the value `BQB1DxEpbF62pk1ps6TzbVna3F...`. The response body is displayed in JSON format, showing details about an album.

```
1   "album_type": "album",
2   "artists": [
3     {
4       "external_urls": {
5         "spotify": "https://open.spotify.com/artist/6ueGR6SWhUJfvEhqkvMsVs"
6       },
7       "href": "https://api.spotify.com/v1/artists/6ueGR6SWhUJfvEhqkvMsVs",
8       "id": "6ueGR6SWhUJfvEhqkvMsVs",
9       "name": "Janelle Monae",
10      "type": "artist",
11      "uri": "spotify:artist:6ueGR6SWhUJfvEhqkvMsVs"
12    }
13  ],
14  "available_markets": [
15    "AD",
16    "AE",
17    "AG",
18    "AL",
19    "AM",
20  ]
```

The screenshot shows the Postman interface with the same GET request. The **Headers** tab is selected, listing several headers including `Authorization` set to `Bearer BQB1DxEpbF62pk1ps6TzbVna3F...`. The response body is identical to the previous screenshot.

```
1   "album_type": "album",
2   "artists": [
3     {
4       "external_urls": {
5         "spotify": "https://open.spotify.com/artist/6ueGR6SWhUJfvEhqkvMsVs"
6       },
7       "href": "https://api.spotify.com/v1/artists/6ueGR6SWhUJfvEhqkvMsVs",
8       "id": "6ueGR6SWhUJfvEhqkvMsVs",
9       "name": "Janelle Monae",
10      "type": "artist",
11      "uri": "spotify:artist:6ueGR6SWhUJfvEhqkvMsVs"
12    }
13  ],
14  "available_markets": [
15    "AD",
16    "AE",
17    "AG",
18    "AL",
19    "AM",
20  ]
```

email: contact@arene.dev

# More advanced usage of Postman.

The screenshot shows the Postman application interface. On the left, the sidebar displays the 'CSSE6400 API Collection' with sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. Under APIs, there are two entries: 'Spotify API' and 'SWAPI (Star Wars API)'. The 'Spotify API' entry has a sub-item 'GET https://api.spotify.com/v1/albums/{id}'. The main workspace shows a GET request to 'https://api.spotify.com/v1/albums/({album\_id})'. The 'Pre-request Script' tab is selected, containing the following JavaScript code:

```
1 pm.sendRequest({
2   url: 'https://accounts.spotify.com/api/token',
3   method: 'POST',
4   header: {
5     'Authorization': 'Basic Yzg0NTc2ZjV1MGU4NGYxNTk1MjV1Y2FmYzNhNjMzMnU0NjMxNzRh0WQ3NDbhNDU0Mj1hNwE3NT1mODY1MjE4ZjY=',
6     'Content-Type': 'application/x-www-form-urlencoded'
7   },
8   body: {
9     mode: 'urlencoded',
10    urlencoded: [
11      { key: 'grant_type', value: 'client_credentials' },
12    ]
13  }
14 }, function (err, res) {
15   pm.globals.set("spotify_access_token", res.json().access_token);
16 });
});
```

The 'Body' tab shows the response structure in JSON format:

```
1 {
2   "album_type": "album",
3   "artists": [
4     {
5       "external_urls": {
6         "spotify": "https://open.spotify.com/artist/6ueGR6SwUhUjfvhqkvMsVs"
7       },
8       "href": "https://api.spotify.com/v1/artists/6ueGR6SwUhUjfvhqkvMsVs",
9       "id": "6ueGR6SwUhUjfvhqkvMsVs"
10     }
11   ]
12 }
```

The status bar at the bottom indicates a successful response: Status: 200 OK Time: 174 ms Size: 34.16 KB Save Response. The bottom right corner contains the email address: **email. contact@larene.dev**.

# What can we "reverse engineer" about uq.edu.au

- Homepage
- Search
- Login

The screenshot shows a web browser window with the following details:

- Address Bar:** uq.edu.au/search/?q=disability&submit=&output=xml\_no\_dtd&client=ws&proxystylesheet=ws
- Page Title:** Search - The University of Queensland
- Header:** THE UNIVERSITY OF QUEENSLAND AUSTRALIA | Search
- Search Bar:** accessibility
- Filter:** All of UQ
- Results:** Web (64900), People, Study, News
- Results List:**
  - Accessibility and study hacks - Library - University of ...**  
https://web.library.uq.edu.au/research-tools-techniques/digital-essentials/accessibility-study-hacks  
Accessibility benefits everyone. Accessibility isn't just about altering existing things to help a specific group of people; it's about changing how we do things on a structural level to remove barriers that could affect anyone. Making websites accessible makes them easier for everyone to use. Accessibility is not just about disability.
  - Accessibility - eLearning - University of Queensland**  
https://elearning.uq.edu.au/guides/accessibility  
What is web accessibility? The web is designed for all people irrespective of their hardware, software, location, language or ability. Under the Disability Discrimination Act 1992, Australian Government agencies are required to ensure information and services are provided in a non-discriminatory accessible manner. All information published by Federal government organisations should be compliant ...
  - Accessibility and study hacks - Library - University of ...**  
https://web.library.uq.edu.au/node/4060/3  
4. Study hacks. Many of the tools specifically designed to improve the accessibility of online content have been widely adopted by many types of users and marketed by companies as general productivity tools. Using these tools can be a great way to hack your studies. Disability-led innovations for the masses (YouTube, 18m35s) demonstrates how ...
  - The Interpretation of Protein Structures: Estimation of ...**  
http://comprobio.chemistry.uq.edu.au/mediawiki/upload/d/d5/Lee1971\_JMolBiol55-379.pdf  
The average change in accessibility for the atoms of all three proteins in going from a hypothetical extended
- Developer Tools (Console Tab):**
  - Log XMLHttpRequests (checked)
  - Eager evaluation (checked)
  - Autocomplete from history (checked)
  - Evaluate triggers user activation (checked)

Logs:

  - ② sent web: uq.edu.au: accessibility to GA main.js?7:486
  - ▼ XHR finished loading: GET "https://www.uq.edu.au/search/search.php? jquery-1.8.0.min.js:2 q=accessibility".
  - send @ jquery-1.8.0.min.js:2
  - ajax @ jquery-1.8.0.min.js:2
  - submitQuery @ main.js?7:508
  - submitCurrent @ main.js?7:48
  - onsubmit @ ?q=disability&submit\_xstylesheet=ws:148

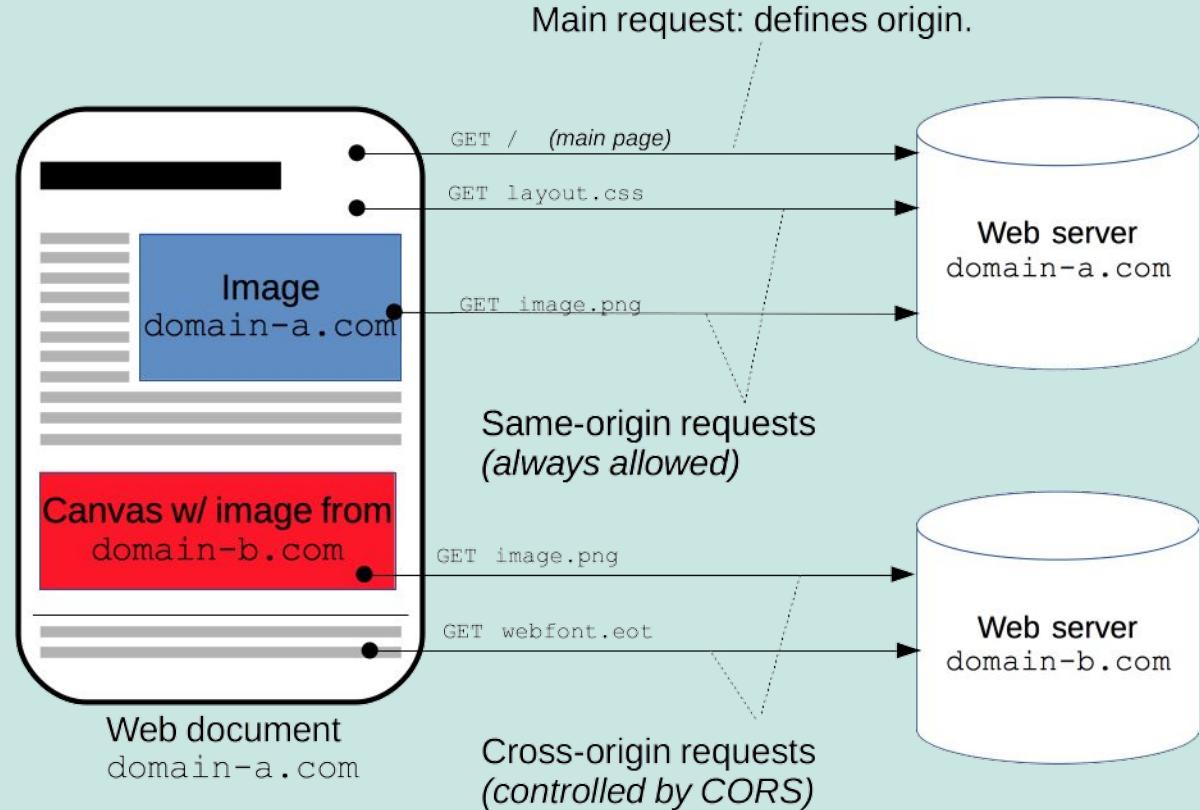
email: contact@larene.dev

# Why do I keep getting CORS errors??

email: [contact@larene.dev](mailto:contact@larene.dev)

# What is CORS?

- A commonly misunderstood concept!
- Limit which websites ("origins") can access your APIs
- Server-side app defines the restrictions. Browser does the restricting.



# Example of CORS

An example of a cross-origin request: the front-end JavaScript code served from `https://domain-a.com` uses `XMLHttpRequest` to make a request for `https://domain-b.com/data.json`.

# Example trying to send a x-origin request from google.

```
> var xhttp = new XMLHttpRequest()
< undefined
> xhttp.open("GET", "https://www.uq.edu.au/search/search.php?q=disability", true);
xhttp.send();
< undefined
▶ XHR failed loading: GET "https://www.uq.edu.au/search/search.php?q=disability". VM501:2
✖ Access to XMLHttpRequest at 'https://www.uq.edu.au/search/search.php?q=disability' from origin 'https://www.google.com' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.
✖ ▶ GET https://www.uq.edu.au/search/search.php?q=disability net::ERR_FAILED 200 VM501:2 ⌂
> :
```

# CORS "errors" may be misleading!

- Sometimes a bug can be mistaken as a CORS error on the browser.
- You can test this by using Postman to hit the endpoint.

Console Elements Network Application Sources > 4 3 ⚙️ ⋮

Preserve log Disable cache No throttling ⚡ ⌂ ⌃ ⌄ ⌅ ⌆ ⌇

Filter Invert Hide data URLs

All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other Has blocked cookies

Blocked Requests 3rd-party requests

20 ms 40 ms 60 ms 80 ms 100 ms

Path	Status	Domain	Type	Initiator	Size	T.	Waterfall
/search/search.php	CORS e...	www.uq...	xhr	VM550:2	0 B	8...	<ul style="list-style-type: none"><li>Open in new tab</li><li>Clear browser cache</li><li>Clear browser cookies</li><li>Copy</li><li>Block request URL</li><li>Block request domain</li><li>Replay XHR</li><li>Sort By</li><li>Header Options</li><li>Save all as HAR with content</li></ul>

Contact@larene.dev

Postman

Home Workspaces API Network Reports Explore

Search Postman

Untitled Request

No Environment

Save Send

Import Request

Untitled Request

GET

Part

Query

Body

Response

Cookies

APIs

Environments

Mock Servers

Monitors

Flows

History

My Workspace

New Import

My first collection

First folder inside collection

Second folder inside collection

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create collection

Import

File Folder Link Raw text Code repository New

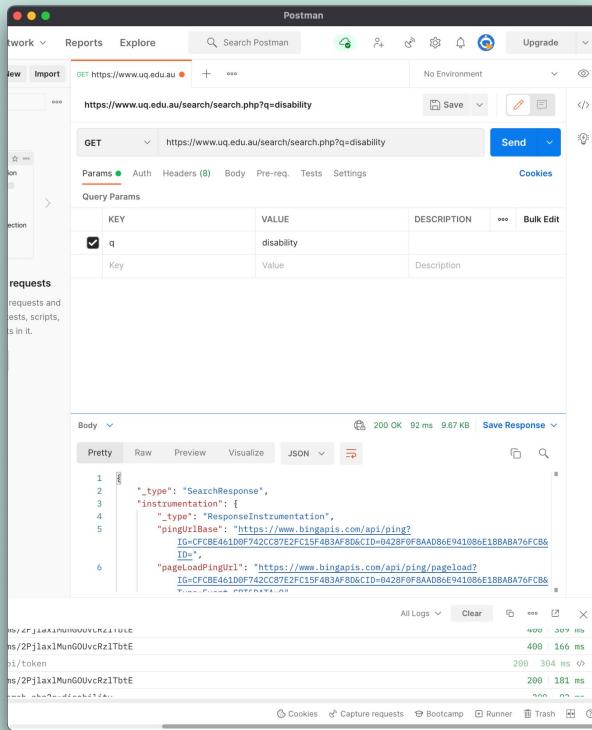
Paste raw text

```
curl 'https://www.uq.edu.au/search/search.php?q=disability' \
-H 'Connection: keep-alive' \
-H 'sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="99", "Google Chrome";v="99"' \
-H 'DNT: 1' \
-H 'sec-ch-ua-mobile: ?0' \
-H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.83 Safari/537.36' \
-H 'sec-ch-ua-platform: "macOS"' \
-H 'Accept: */*' \
-H 'Origin: https://www.google.com' \
-H 'Sec-Fetch-Site: cross-site' \
-H 'Sec-Fetch-Mode: cors' \
-H 'Sec-Fetch-Dest: empty' \
-H 'Referer: https://www.google.com/' \
-H 'Accept-Language: en-US,en;q=0.9,ru;q=0.8,uk;q=0.7,de;q=0.6,es;q=0.5,fr;q=0.4,pt;q=0.3,ro;q=0.2,pl;q=0.1' \
-H 'Sec-Fetch-Dest: document' \
-H 'Sec-Fetch-Mode: navigate' \
-H 'Sec-Fetch-Site: none' \
-H 'Sec-Fetch-User: ?1' \
-H 'TE: trailers'
```

Continue

Enter the URL and click Send to get a response.

# No CORS restrictions to worry about in Postman!



email: contact@larene.dev

# Why is my website not online?

email: [contact@larene.dev](mailto:contact@larene.dev)

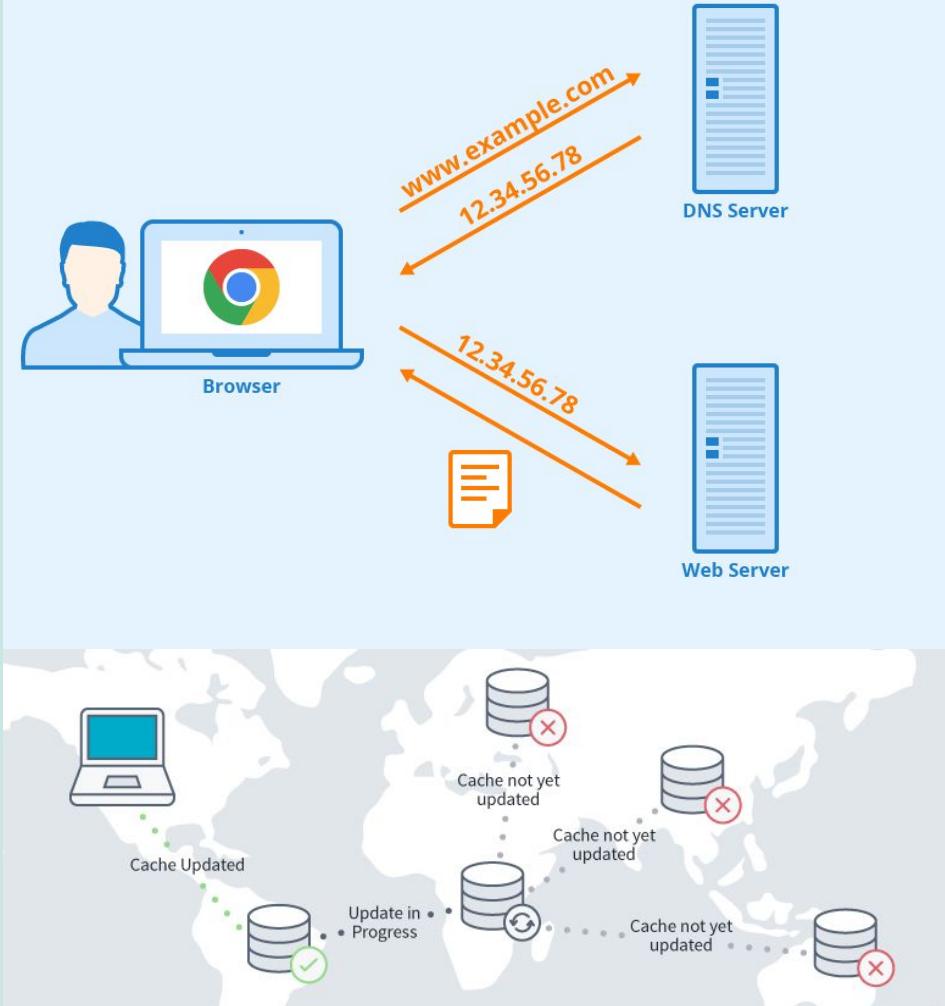
# What is DNS?

A Domain Name System (DNS) is a **decentralised address book** for domains and their respective IP addresses.

When you change DNS settings, it **takes a while** to roll out the changes all over the world.

Source:

<https://glowfroghosting.com/dns-propagation-how-long-do-dns-changes-take/>



# How to check DNS

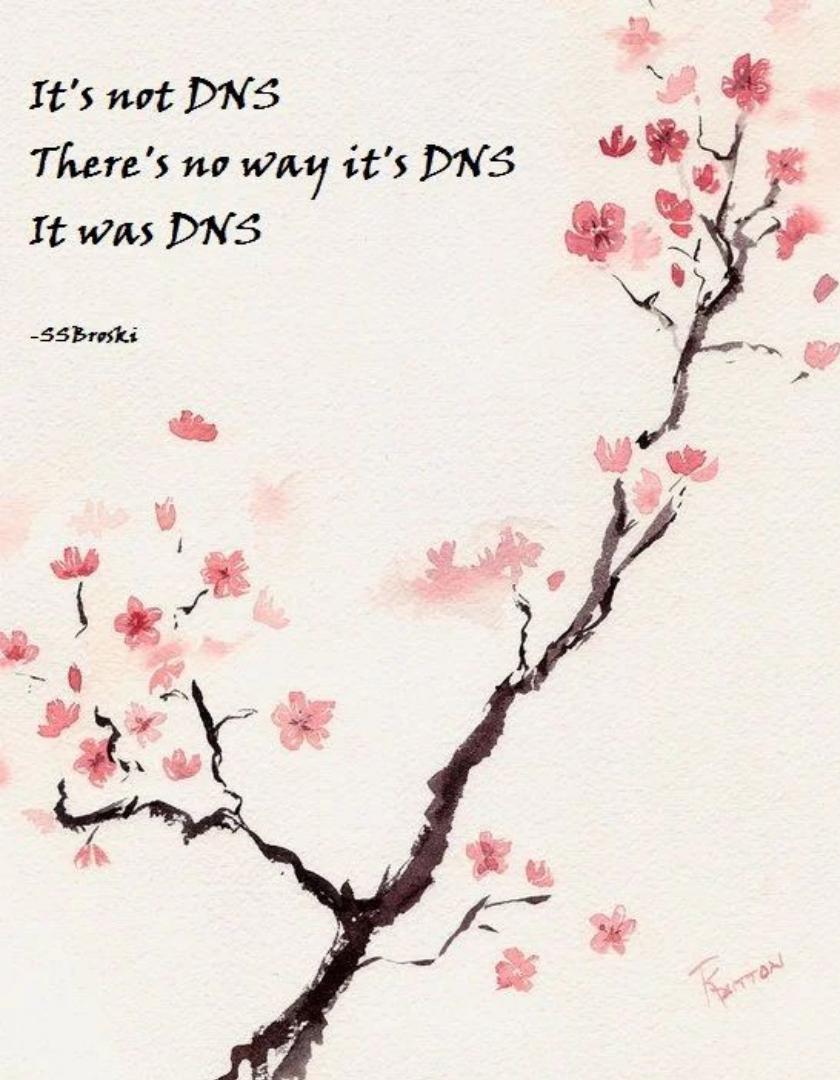
- "nslookup" command in the command line
- [Networks Tools: DNS, IP, Email](#)
- [DNS Propagation Checker](#)
- [IP Address Lookup](#)

*It's not DNS*

*There's no way it's DNS*

*It was DNS*

-SSBroski



Source: [A haiku about DNS - It's not DNS. There is no way it's... - nixCraft](#)

**email:** [contact@larene.dev](mailto:contact@larene.dev)

# Summary

- Oh no, our app, it's broken!
  - Browser Developer Tools
- Why are my changes not showing up?
  - How and where to reset cache
- Exploring and testing APIs
  - Using Postman
- Why do I keep getting CORS errors?
  - Browser Developer Tools + Postman
- Why is my website not online?
  - nslookup
  - MX Toolbox
  - DNS Checker
  - IP Lookup

# Thank you!

email: [contact@larene.dev](mailto:contact@larene.dev)