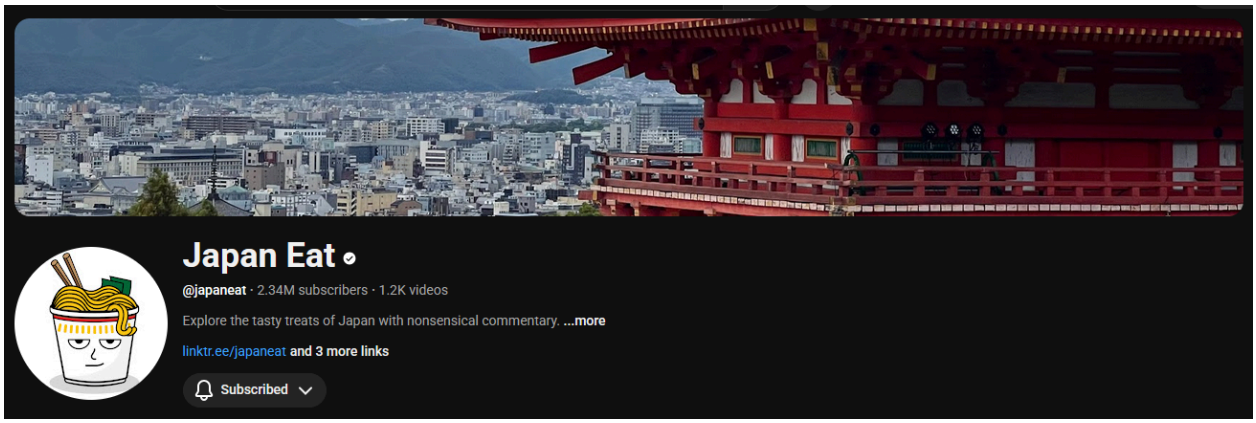
Emergency Blog: My 15 Minute Brainrot Project



So honestly I didn't have a blog idea today. I was debating talking about my Master's degree retrospective, how my TFT gameplay is impeccable, and maybe even things I wanna do in the coming months. However at 9pm central time (3 hours before the "deadline"), I got nerdsniped by Youtube's refusal to allow autoplay on their Shorts tab. As a result, instead of working on my semester project I wrote up a quick script to capture the API request, read the response, and set a timer to press the down arrow key after the timer.

Once I had the general framework mapped out I did the initial scaffolding with Gemini and then fixed all the bugs while testing on one specific channel. Within 15 minutes I had an MVP script that did what I sought out to do. For those interested I'll put the code down below in plain text...

```
(function() {
    // 1. Store the original fetch function so we don't break the website
    const originalFetch = window.fetch;
    const URL_TO_LISTEN_FOR = 'player?prettyPrint=false';
    const CUSTOM_TIME_PADDING_MS = 500;

    // 2. Override fetch with our custom logic
    window.fetch = async function(...args) {
        // 1. Capture Start Time
        const startTime = performance.now();
        // Allow the request to complete normally
        const response = await originalFetch.apply(this, args);
        // 2. Capture End Time & Calculate Duration
        const endTime = performance.now();
```

```
        const networkLatency = (endTime - startTime).toFixed(2); // Duration in milliseconds
        // console.log(args[0]?.url) // Debug to find where the player URL is located
        // Check if the URL string contains the specific endpoint
        // Since args[0] is a Response object base null checks and coalesce SHOULD be enough
        if (args[0] && args[0]?.url.includes(URL_TO_LISTEN_FOR)) {

            // We must clone the response because the response body can only be read once
            const responseClone = response.clone();
            responseClone.json().then(data => {
                // console.log(data) // Just a sanity check to make sure we have the right object
                // 3. Navigate the object to find lengthSeconds
                if (data.videoDetails && data.videoDetails.lengthSeconds) {
                    const TIMER = parseInt(data.videoDetails.lengthSeconds, 10);
                    console.log(`[Script] Target request found.`);
                    console.log(`[Script] Video length: ${TIMER} seconds.`);
                    console.log(`[Script] Latency of Request: ${networkLatency}ms.`)
                    console.log(`[Script] Timer started...`);

                    // 4. Set the sleep timer
                    setTimeout(() => {
                        triggerKeyPress();
                    }, (TIMER * 1000) - networkLatency); // Convert seconds to milliseconds
                }
            }).catch(err => {
                // Ignore JSON parsing errors for other request types
            });
        }
        return response;
    };

    // Helper function to simulate the space bar press
    function triggerKeyPress() {
        console.log("[Script] Timer finished. Pressing Space Bar.");
        // We target the movie player specifically, or fallback to body
        const target = document.querySelector('.html5-video-player') || document.body;

      const keyEvent = new KeyboardEvent('keydown', {
            bubbles: true,
            cancelable: true,
            key: ' ',
            code: 'ArrowDown',
            keyCode: 'ArrowDown',
            charCode: 40
        });
        target.dispatchEvent(keyEvent);
    }
    console.log("[Script] Fetch interceptor is now active. Reload the page or click a video.");
})();
```

Basically I just go into the browser's developer console and execute that code block. From there I let my brain turn into mush as the last semblance of action has been wrested away…

Anyways, have a good day. I have some Youtube shorts to watch.

P.S. Next week is going to be another longer blog 👀