COMP 266: Unit 7

External Data Sources.

Richard Sullivan Andison

2026, February

Unit Review

External data sources can be used in a variety of ways to enhance or add to a website's functionalities or appearance. CSS styling sheets can be loaded from external sources as well as images and libraries, such as jQuery; for some sites, utilizing external sources can be a means of ensuring features remain up-to-date without requiring intervention at every stage; furthermore, utilizing online-resources removes the need to keep and maintain local copies that can take up space in a project. For the New 3DS Library, external data includes the catalogue JSON file and the current collection of book texts that having been provided by ProjectGutenberg.org and ProjectGutenberg.ca. In earlier stages of development, the plan was to build a catalogue that loaded literature via web-scraping book links from the Gutenberg sites (such as the top 100) and allow users to request custom links to literature (this would allow any form of HTML literature to be formatted and read for 3DS systems online), but this idea was scrapped due to the the limitations in the current site-hosting system (the GitHub.io Cross-Origin Resource Sharing (CORS)) and complexity. Instead, the current system for implementing external resources involves uploading literature (and the catalogue JSON) to the host GitHub repo before sending a request for the uploaded resources; while this is a fairly roundabout way of accessing local data, but it allows for practice in JavaScript web requests.

Online implementations evolved in sevaral stages. The earliest stage was simply finding a means for requesting data that worked and attempting to send and display raw text on the site's page. The first method tested was JavaScript's fetch function, but this method did not work for the 3DS. The next method tested was the older AJAX (Asynchronous JavaScript and XML) means for receiving data [1], [2]. While the AJAX method worked, there was significant difficulty in navigating and developing a proper understanding for callback functions and scoping in JavaScript [3]. After a basic framework for recieving data was created, formatting literature was improved by switching from raw text to an HTML format; this new feature required new functions for parsing HTML by paragraph tags and inserting HTML into existing pages (.indexOf()); furthermore, much of the literature required some level of manual reformatting to ensure proper display on 3DS system (such as stripping existing text styling). In addition to providing better formatted text, HTML literature allowed for the inclusion of photographs; all photos required reformatting so they would fit within the 3DS's screen dimentsions and not take too long to load (the New 3DS is limited to 2.1Mb/s connection speeds) [3]. After a function for recieving and formatting HTML formatted literature was completed, another function was designed to request and read from a CSV file for populating the catalogue page menu. The CSV function was similar, but parsed the raw text using JavaScript's .split() method to parse each line method a regex expression to parse each data entry; this was somewhat complicated, but allowed for some practice in both methods [5]. The CSV fetch function was later

changed to receive and parse JSON files instead to simplify code and utilize JavaScripts native data types. Next, the HTML and CSV functions were again simplified using jQuery before finally being merged into a single “get” function that worked in a way similar to the non-functioning fetch function. At the time of writing, the new fetch function takes a url, data type, and function; a jQuery AJAX function is called with a set to call GET request for a data type of the specified parameter in where the passed function is called on the data received back. With new fetch get function, a programmer needs to only concern themselves with functions that handle the data passed back. The fetch function was later changed from “get” to “DSFetch()” to avoid naming confusion (get is already used in JavaScript as a keyword) [6]. Therefore the functions evolved from getText and getCSV to getText and getJSON to simply get and finally DSFetch.

Some areas related to external data would include jQuery, graphics, and GitHub. While jQuery could be implemented by linking to the official site resource, I decided to store a copy of the latest version (4.0.0) as it reduces the need to load from a separate external source. Furthermore, because the 3DS browser is fairly outdated, it is unlikely that any future versions of jQuery will add any additional features or improved functionality for the 3DS browser (the latest version removed support for browsers older than Internet Explorer 10) [7]. The majority of additional graphics (such as the candle gif, rocking avocado, and sun) all originated from Pixabay and are free to use; that being said, future improvements could involve citing specific authors to ensure

they get recognition for their art; currently the GitHub repo for major changes has credits that specifies each artists, but a table could be implemented on the DS site credits page in the future. Finally, GitHub was used throughout the entirety as a means of hosting the website and keeping track of all changes. In general, GitHub's CLI was used to perform a basic add, commit, push cycle; some commits were restored or previous versions were pulled to revert changes at times. Future deployments could see a new repo with a stable and testing branch so a functioning version of the site can remain while testing is done on the other (this could also introduce the opportunity to practice merging and more complicated Git commands).

In conclusion, the New 3DS Library utilizes several online-resources and external data sources to acquire literature, add functionalities (such as jQuery), graphics (Pixabay), and allow for tracking site changes and hosting (GitHub). Implementing a means for requesting and parsing catalogue and book data took the most time to complete and evolved through multiple stages. For external libraries, jQuery provided a means of simplifying and abstracting away many elements of the original JavaScript code. While no external CSS elements where currently utilized, the site does utilize several graphics taken from Pixabay to add to themes. GitHub was used for site hosting and for tracking changes.

Resources Used

[1] “AJAX The XMLHttpRequest Object,” *www.w3schools.com*.

https://www.w3schools.com/js/js_ajax_http.asp

[2] “XMLHttpRequest,” MDN Web Docs, May 10, 2019. https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest

[3] “JavaScript Callbacks,” www.w3schools.com.

https://www.w3schools.com/js/js_callback.asp

[4] Thalantyr1216, “3DS Download Speeds,” *Gamespot.com*, May 12, 2020.

https://gamefaqs.gamespot.com/boards/997614-nintendo-3ds/78680597 (accessed Feb. 25, 2026).

[5] Firas Dib, “Regex101 - online regex editor and debugger,” @regex101, 2019.

https://regex101.com/

[6] “get - JavaScript | MDN,” *Mozilla.org*, Oct. 31, 2025. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/get (accessed Feb. 25, 2026).

[7] jQuery Foundation - jquery.org, “jQuery 4.0.0 | Official jQuery Blog,” *Jquery.com*, Jan. 18, 2026. https://blog.jquery.com/2026/01/17/jquery-4-0-0/ (accessed Feb. 20, 2026).