# 6.5.8 Reflection

Congratulations on deploying your first application that uses a server-side API! Remember that you now have the power to interact with the ever-growing number of APIs out there.

In this lesson, you accomplished the following:

- Learned about the options available for GitHub API endpoints.
- Used multiple parameters in a query string.
- Dynamically updated an API call using HTML attributes.

This is already impressive, especially considering everything else we've touched on while working on the Git it Done app. Let's review what you've done throughout the module:

- Learned that a server-side API is an interface for accessing another website's data.
- Familiarized yourself with how to read server-side API documentation.
- Used the browser's Fetch API to make requests to a server-side API.

- Gleaned additional information about requests being made, using the DevTools Network tab.
- Captured user input from the browser to create dynamic HTTP requests.
- Learned how to use an HTTP request's response to display data to a user.
- Learned how to handle errors that may occur when working with server-side APIs.
- Learned that request and response headers contain additional information about the request separate from the data itself. For example, GitHub's Link header lets you know there are more pages of results to request.
- Leveraged another GitHub API endpoint to request more specific data. This endpoint also used an optional `?` string to change how results are sorted.
- Used query parameters to pass information from one page to another.
- Used the `document.location` object to read from a URL query string.

Not only did you learn essential concepts in server-side APIs while making this app, but you also made it easier to browse GitHub. Your colleague Amiko plans to use the app now to contribute to HTML and CSS-related projects.

On that note, why don't you contribute to open source JavaScript projects to get a taste of the collaboration that's so integral to web development? Working on these projects will strengthen both your technical skills and your communication skills. As you tackle new issues, you'll interact with a community of developers and integrate with a workflow like the one you'll experience on the job.

Not to mention, contributions to open source projects look great on your resume. Through those contributions, employers gain insight into how you work with other people and how well you write code. Even though it might feel intimidating to make your first contribution, start small and don't underestimate what you can learn from a single contribution!