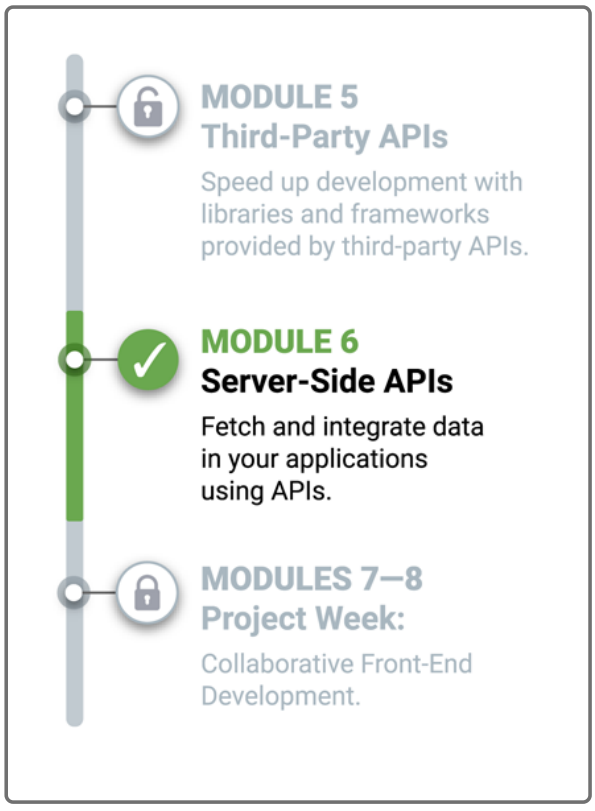
6 Roadmap

Looking Ahead: This Week's Challenge

By the end of this week, you'll complete your weekly Challenge assignment where you will build a weather dashboard application that allows users to search for a city and find its current and forecasted weather conditions.



- Use the fetch API to send requests to a server-side API
- Receive and parse data in the JSON format
- Use returned data to dynamically generate HTML
- Handle response codes and metadata, including headers, status, and URL

The online lessons in this week's module will teach you the skills you need to succeed in this Challenge. You'll learn core server-side API skills by

developing an application that helps users search GitHub by username or topic, and display matching projects along with the number of open issues they have.

What You Will Learn

By completing this module, you will learn how to:

- Explain the difference between a client-side API and a server-side API
- Explain the client-server model and request-response pattern
- Explain and implement the differences between HTTP GET requests using XMLHTTPRequest, jQuery AJAX, and the fetch API
- Explain HTTP response codes and handle response metadata with fetch API
- Parse JSON to dynamically generate HTML
- Explain the benefits and challenges of working with asynchronous JavaScript
- Explain and implement query string parameters

Weekly Tip: Elaborate on Complex Ideas to Understand Them

As a web developer, you'll encounter plenty of complex concepts and techniques throughout your career. This might feel intimidating, but rest assured that your brain is built to accommodate intricate, nuanced ideas. In fact, the human brain has as many neurons as there are stars in this galaxy and is one of the most complex systems in the universe!

This interconnectedness helps you learn new complex concepts. For example, have you ever heard of a liger? A liger is a cross between a lion and a tiger. Can you picture that in your mind? Even if you've never heard of a liger before, you can probably guess what it looks like by connecting what you already know about what lions and tigers look like to something new. (As you probably guessed, ligers look a bit like small tigers with faded stripes.)

This week, use one of the following techniques to integrate your existing coding knowledge with new knowledge or ideas from the lessons:

Make an Analogy

As you learn something new, compare it with concepts from your everyday life or from other subjects. Take this information: `while` loops repeat as long as a certain Boolean condition is met. You might choose to think of `while` loops like a crosswalk. As long as the light is green (or true), people will keep crossing the street. Only when the light turns red (or false) do they stop.

Visualize It

Brains remember images and places more easily than words. If you connect a new idea to a meaningful visual or a specific place, you'll situate the new knowledge and be more likely to remember it later.

Let's take this concept from the last module: If you close the browser, the data in `sessionStorage` will disappear, but the data in `localStorage` will persist. You might find it useful to visualize little safe-deposit boxes inside the computer itself (`localStorage`) or inside the browser (`sessionStorage`). When you close the browser, `sessionStorage` gets emptied, but `localStorage` stays full.

Choose your analogies and visuals carefully—it can be hard to unlearn an analogy if it leads you down the wrong path.

Ask Yourself Questions

You can also boost comprehension and memory by asking yourself “how” and “why” questions.

For example, you could look at a working website’s source code and ask yourself questions like, “Why does this work?” or “Why did they choose this particular HTML function?” Then answer the questions to the best of your ability based on what you know.

If you start to feel overwhelmed by the complexity of this boot camp, rest assured that your brain is well equipped to handle difficult concepts. The key is to leverage what you’ve learned so far and make connections where possible.

Planning Your Schedule

This module consists of five lessons. Here's an estimate of where you should be as you work through this module:

- By the time you reach your Virtual Class (Recommended), you should have completed at least Lesson 1. Push yourself to start Lesson 2 by our first meeting.
- By the time you reach your Virtual Class (Required), you should have completed at least Lesson 2 and half of Lesson 3.
- Start looking at the Challenge assignment early in the week. You can begin working on the Challenge at any point.