JSON Server

- **json-server** is an NPM package that spins up a mock backend for our front end application.

- We configure a **json** file with our desired endpoints and their responses. In our app, we created a **/jobs** endpoint that returns an array of job objects.

Axios

- The **axios** library makes HTTP requests. We use the **axios.get** method to make a GET request to an endpoint for data.

- Within a component, a common place to fetch data is the **mounted** lifecycle hook. The async request may take some time to resolve, and we don’t want to delay component render.

Factory Functions

- The response can overwrite a **data** property, which forces Vue to re-render the template with the latest data.

- We can use factory functions to generate complex objects for our tests. We can design them in such a way that our tests can specify the unique properties they care about.

Mocking out Libraries

- We don’t *want* our component unit tests to make an actual API request. It’s slow, prone to error, and complex.

- Rather, we can instruct Vitest to *mock* out a dependency (like Axios). Vitest will replace all of an object’s methods with mock functions.

Mocking out Libraries

- We can customize how our mock functions operate, such as setting their return values or resolved values. By default, Vitest functions return **undefined**.

- **Key Takeway:** Create the simplest, most lightweight object that will allow us to test the component in isolation.

Vue Testing Library Async Query Methods

- The **find** family of methods from Vue Testing Library return a Promise. This is helpful when we need to wait for a component to rerender due to some asynchronous operation completing.

- The method names will be similar (**getByRole** for sync versus **findByRole** for async)