

Vue.js: GET AND DISPLAY JSON DATA [FETCH VS AXIOS]

Introduction

One of the fundamental tasks of any web application is to communicate with servers through the HTTP protocol. This can be easily achieved using Fetch or Axios. Fetch and Axios are very similar in functionality .

Some developers prefer Axios over built-in APIs for its ease of use. Yet still the Fetch API is perfectly capable of reproducing the key features of Axios.

Fetch API

The Fetch API is a standard API for making HTTP requests on the browser. It's a great alternative to the old XMLHttpRequest constructor for making requests. It supports all kinds of requests, including GET, POST, PUT, PATCH, DELETE, and OPTIONS, which is what most people need.

- ❖ To make a request with the Fetch API, we don't have to do anything. All we have to do is to make the request directly with the fetch object.
 - ❖ We make a simple GET request from an API and then convert the data from JSON to a JavaScript object with the json() method.
 - ❖ Then we display the object's data directly on the template. We can also process response bodies in other formats with the Fetch API, including plain text and binary data.
 - ❖ Like most HTTP clients, we can send request headers and bodies with the Fetch API.
-

Axios

Axios is a popular HTTP client that works on both browser and Node.js apps. It is a Javascript library used to make HTTP requests from node.js or XMLHttpRequests from the browser . It can be used to intercept HTTP requests and responses and enables client-side protection against XSRF. It also has the ability to cancel requests.

- ❖ We install it and use it to make simple GET requests.
- ❖ we call the `axios.get` method with the URL to make the request.
- ❖ Then we assign the response data to an object.
- ❖ To make a POST request with a request body, we can use the `axios.post` method. We make the POST request with the `axios.post` method with the request body in the second argument.
- ❖ Then we get back the response data by getting the `data` property from the resulting response.

WHICH SHOULD YOU USE:

The most common way for frontend programs to communicate with servers is through the HTTP protocol. As with Fetch, Axios is promise-based. However, it provides a more powerful and flexible feature set.

One feature that it has over `.fetch()` is that it performs automatic transforms of JSON data. If you use `.fetch()` there is a two-step process when handling JSON data. The first is to make the actual request and then the second is to call the `.json()` method on the response.

So by using axios you can cut out the middle step of passing the results of the http request to the `.json()` method. Axios just returns the data object you would expect.

To put it in simple terms :

Axios over the native Fetch API include

- Request and response interception
- Streamlined error handling
- Protection against XSRF
- Support for upload progress
- Response timeout
- The ability to cancel requests
- Support for older browsers
- Automatic JSON data transformation

If you're working on multiple requests, you'll find that Fetch requires you to write more code than Axios, even when taking into consideration the setup needed for it. Therefore, for simple requests, Fetch API and Axios are quite the same. However, for more complex requests, Axios is better as it allows you to configure multiple requests in one place.