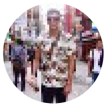
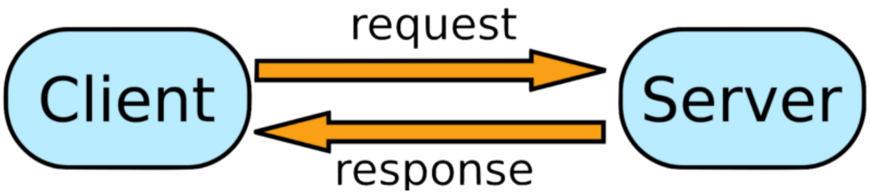
HTTP Requests Compared: Why Axios Is Better Than Node-Fetch (Automatic Transformations, More Secure, Can Handle Errors Better, Interceptor Support, And More Browser Friendly)

By Jeff Lewis



Jeff Lewis Follow

Jul 19, 2018 · 5 min read



What Are They?

Axios and node-fetch both promise-based request libraries to perform HTTP requests for retrieving, posting, deleting, and modifying data from APIs. What's an HTTP request? An HTTP request is use to send and recieve data between the client and server, in which you would use a HTTP method (GET, POST, PUT, DELETE, etc.). These HTTP requests can be made from the client or server, depending on your project.

Why Do We Use Them?

Say you wanted to connnect to a GraphQL or REST API and you wanted to send data to and the API to your web application. You would need to do this as an HTTP request using a HTTP method (GET, POST, PUT, etc.). When just diving into APIs, most people tend to start off using the SWAPI (Star Wars API) or Pokéapi (Pokemon API) because they are free to use and an API key is not required. Examples below will be using the Pokéapi.

What's The Difference?

You're probably just looking for the answer of which one is better. **Axios is better (NPM Link)** and listed below are a few reasons why. First, let's look at a sample GET and POST request using ES6 arrow functions so we have a basis what an Axios GET and POST request looks like.

Axios GET/POST Examples

Axios HTTP Request (GET):

```
axios.get('http://pokeapi.co/api/v2/pokemon/')
  .then((response) => {
    return response;
  })
  .catch((error) => {
    console.log(error);
  });
}
```

Axios HTTP Request (POST):

```
axios.post('http://pokeapi.co/api/v2/pokemon/', {
  pokemon: 'Pikachu',
  })
  .then((response) => {
    return response;
  })
  .catch((error) => {
    console.log(error);
  });
```

1. Axios Performs Automatic Transformations Of JSON Data.

Transformers allow automatic transformations to convert the response to JSON data. Us programmers aren't lazy, we just know how to do things the correct way. This means one less step to perform, one less thing to go wrong, and a reduced file size for more drive space on your computer for that "1998 Taxes" folder (Relax, the last part is sarcasm). The bolded line of code is the extra piece of code node-fetch requires.

Pokéapi GET Request (Node-Fetch):

```
fetch('http://pokeapi.co/api/v2/pokemon/')
  .then((response) => response.json())
  .then((data) => return data)
```

Pokéapi GET Request (Axios):

```
axios.get('http://pokeapi.co/api/v2/pokemon/')
  .then((response) => return response)
```

2. Axios Is More Secure, Featuring Built In Cross Site Forgery (XSRF) Protection

XSRF stands for Cross Site Forgery. What is Cross Site Forgery? XSRF can be explained in this scenario. Say you're on your computer and you log into your bank account, in which your computer receives a session token. You then receive an email cleverly disguised email from a hacker. The email contains a link, in which you click on the link and it runs some code on your computer. It then connects to your bank account that you were previously logged in with using that session token you received from the bank, using your computer as an authorization middleman. OMERGHDDDDD all of my fetch code is going to be hacked! There's a very slim chance since website security has strengthened over the years, but it's always better to be on the safe side.

3. Axios Has Better Error Handling

One of the more popular benefits of Axios has is better error handling. Nothings worse trying to debug a tricky or a poorly handled error. When your backend returns the status code "500 Internal Server Error," fetch will treat it as the same as status code "200 OK." Say for example you forgot to add that extra step of transforming your data to JSON:

Example #1 (Your Code):

```
fetch('http://pokeapi.co/api/v2/pokemon/')
  // Missing the JSON transformation line
  .then((data) => return data)
```

Example #1 (Web Browser Console Response):

```
body: (...)
bodyUsed: false
```

```
headers: Headers
ok: true
status: 200
statusText: "OK"
type: "cors"
url: "http://pokeapi.co/api/v2/pokemon/"
```

You received the status "200 OK," but this is not the JSON data you were looking for. The response from the server letting you know that your request went through, but that doesn't really help you.

4. Axios Has Interceptors

Interceptors allow you to run code or modify the request and/or response before the request and/or response has started. Axios is supports promises and has the ability to run async operations before request is made or before the promises of .then or .catch are executed. For the first example, we use a config.

Pokéapi Config Request Example:

```
axios({
  method: 'get',
  url: 'http://pokeapi.co/api/v2/pokemon/',
  data: {
    pokemonNumber: '1',
  }
});
```

Pokéapi (Add Interceptor To The Request Data):

```
// Add a request interceptor
axios.interceptors.request.use((config) => {
  // Your Interceptor code to do something with the request
data


  // Return config
  return config;
  },
  (error) => {
  // Your Interceptor code to do something with request
error



  // Return error
  return Promise.reject(error);
  }
);
```

Pokéapi (Add Interceptor To The Response Data):

```
axios.interceptors.response.use((response) => {
  // Your Interceptor code to do something with the response
data

  // Return response data
  return response;
  },
  (error) => {
  // Your Interceptor code to do something with response
error

  // Return error
  return Promise.reject(error);
  }
);
```

5. Axios Has The Ability To Monitor POST Request Progress

Say you have a large request that would take longer than normal. You have the promise .catch to send the error, but nothings happened. My god, your gorgeous creation of code is crumbling. I'm kidding. It's probably just that your request is still in progress. Fetch does not provide any way to get notified of the request progress, but axios does. Here's an POST request example:

```
axios.post('http://pokeapi.co/api/v2/pokemon/', data, {
  onUploadProgress: ({ total, loaded }) => {

    // Update progress

  }),
})
```

6. Axios Has A Wider Range Of Supported Browsers

Both HTTP request libraries support Google Chrome, Safari, Opera, and Edge, but his is where axios comes out on top. Axios supports Internet Explorer verison 8 and newer, allowing your web application to be compatible with a wider range of browsers.

The disussion of how your web application is going to be used and who's going to be using it is necessary. If you're developing a web application for a business, a place where computer equipment and software updates are usually neglected along with that raise you've been asking for, there's a good chance someone will be using Internet Explorer.

You're All Done, So Go GET Em, Tiger

And that's it! You now are more informed further on HTTP requests and your knowledge of Axios and what it is capable has been enhanced.

No one's perfect. If you've found any errors, want to suggest enhancements, or expand on a topic, please feel free to send me a message. I will be sure to include any enhancements or correct any issues.