Fetch and Axios

`fetch` and `axios` are both JavaScript libraries that are commonly used for making HTTP requests in web applications. While they serve a similar purpose, there are some differences between them:

1. API Design: `fetch` is a native browser API introduced in ES6 (ECMAScript 2015) that provides a low-level interface for making HTTP requests. It returns a Promise that resolves to the response, which requires additional handling to parse and process. On the other hand, `axios` is a popular third-party library that provides a more convenient and feature-rich API for making HTTP requests. It also returns a Promise-based response but includes built-in JSON parsing and supports various request and response interceptors.

2**. Browser Support:** `fetch` is supported by modern browsers, but it may require polyfills or additional configuration to work in older browsers. `axios`, being a standalone library, has better compatibility and works consistently across different browsers.

3. **Request and Response Handling:** `fetch` uses the `Response` and `Request` objects to handle requests and responses, while `axios` provides a higher-level abstraction. `axios` simplifies request configuration by allowing you to pass parameters directly through method arguments, whereas `fetch` requires manual construction of headers, query strings, and other request details.

4. Interceptors: `axios` offers an interceptor feature that allows you to intercept and manipulate HTTP requests and responses globally or at the request level. Interceptors can be useful for adding common headers, handling errors, or modifying the response data. `fetch` does not have built-in interceptors, so you would need to handle these functionalities manually.

5**. Cancellation:** `axios` provides cancellation tokens, which allow you to cancel ongoing requests. This can be helpful when you need to abort requests that are no longer needed. `fetch` does not have built-in cancellation support, although you can achieve a similar effect by using the `AbortController` interface.

6. Size: If file size is a concern, `fetch` is smaller in size compared to `axios` since it is a native browser API. `axios` includes additional functionality, making it slightly larger.

Overall, the choice between `fetch` and `axios` depends on your specific needs and preferences. If you require a simple and lightweight solution or have compatibility concerns, `fetch` might be a good option. On the other hand, if you need a more feature-rich library with built-in JSON parsing, interceptors, cancellation support, and wider browser compatibility, `axios` is a popular and reliable choice.