Fetch

vs

Axios

Which to Choose for API Calls?

When building an application, handling API requests is a crucial aspect of integrating with backend services.

Two popular options for making HTTP requests are the built-in fetch API and the third-party library axios.

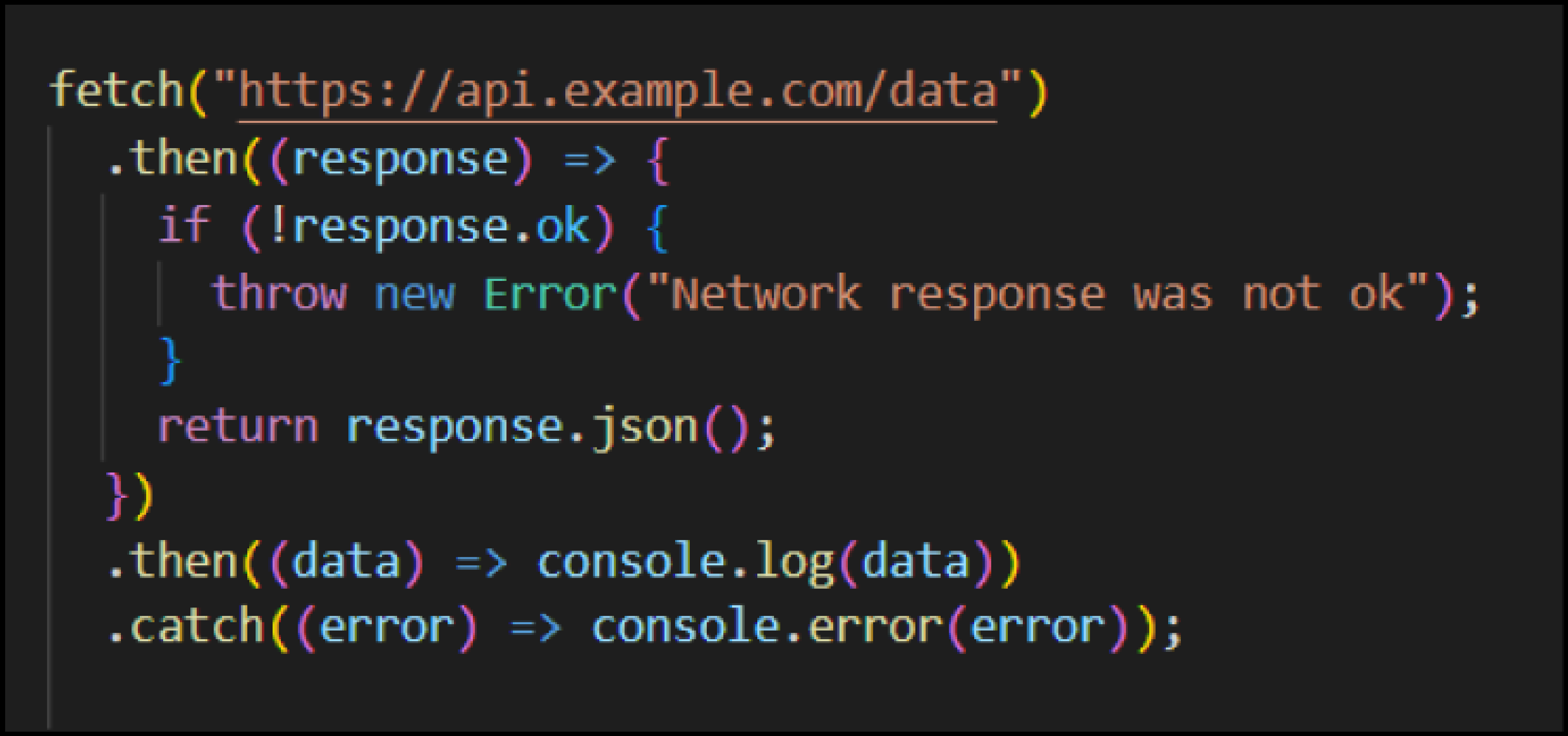
Both have their strengths and weaknesses, and choosing the right one can impact your application's performance and ease of development.

In this post, we'll compare fetch and axios to help you make an informed decision.

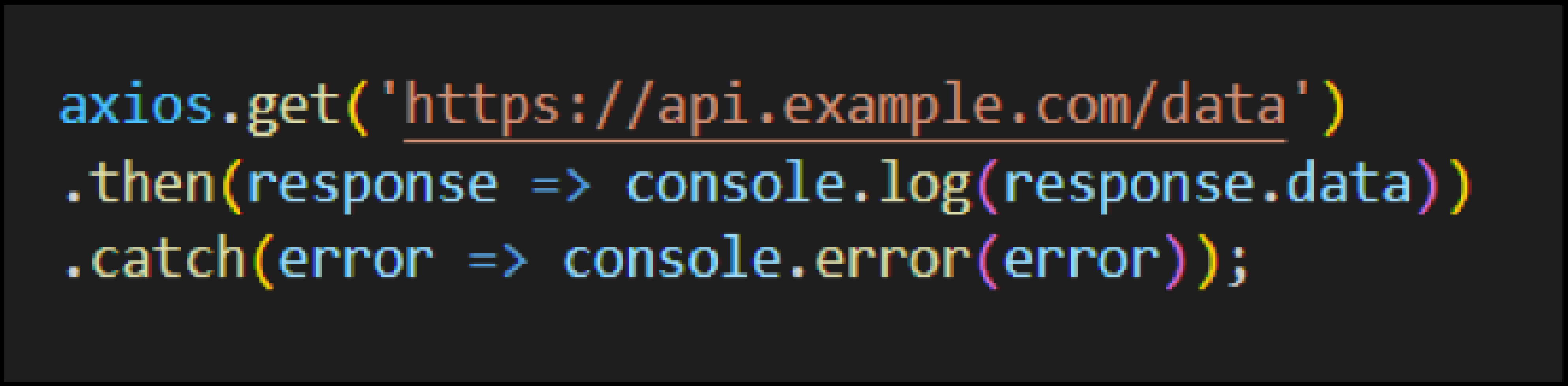
Fetch: fetch is a native JavaScript function that provides a modern, promise-based API for making HTTP requests. It's built into the browser and doesn't require any additional libraries.

Axios: axios is a promise-based HTTP client for the browser and Node.js. It offers a more feature-rich API compared to fetch and is widely used in React applications.

Example Usage



```
fetch("https://api.example.com/data")
  .then((response) => {
    if (!response.ok) {
      throw new Error("Network response was not ok");
    }
    return response.json();
  })
  .then((data) => console.log(data))
  .catch((error) => console.error(error));
```



```
axios.get('https://api.example.com/data')
.then(response => console.log(response.data))
.catch(error => console.error(error));
```

Error Handling

Fetch: Requires manual handling of HTTP errors. fetch only rejects the promise on network errors, not on HTTP errors like 404 or 500. You need to check response.ok and handle errors accordingly.

Axios: Automatically rejects the promise for HTTP errors. You get a more detailed error object with response, request, and message properties.

JSON Data Transformation

Fetch: Requires explicit conversion to JSON using response.json(), which returns a promise.

Axios: Automatically transforms JSON data, so you can access response.data directly.

Conclusion

Both fetch and axios are capable tools for making API requests in React applications. Your choice depends on your specific needs:

- Choose fetch if you prefer using a built-in, lightweight solution and are comfortable handling HTTP errors and request/response transformations manually.
- Choose axios if you need a feature-rich library with built-in support for interceptors, automatic JSON transformation, and request cancellation.

Ultimately, both tools are effective, and the best choice will depend on your project's requirements and your development preferences.