

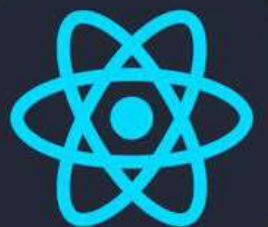
# Axios

## VS

# Fetch API



@developer.mohit



# Usability

## Axios

- Axios is a **third**-party library that needs to be installed (*npm install axios*). It offers a simpler and more concise syntax for making HTTP requests.
- Axios **automatically** handles JSON conversion and provides better error handling out of the box.

## Fetch API

- Fetch is a **built-in** browser API, which means there's no need to install any additional libraries. It uses promises and has a more modern syntax.
- However, Fetch's syntax can become **verbose** when handling common use cases like JSON parsing and error handling.



@developer.mohit

# Syntax

## Axios

```
import axios from 'axios';

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



@developer.mohit

## Fetch API

```
fetch('https://api.com/data')
  .then(response => {
    if (!response.ok) {
      throw new Error('Network
        Error');
    }
    return response.json();
  })
  .then(data => console.log(data))
  .catch(error =>
    console.error(error)
  );
```

# Response Handling

## Axios

- Axios automatically **rejects** promises for HTTP errors, making error handling easier and more intuitive.
- Axios response data is available directly in **response.data**, so no need for additional processing.

## Fetch API

- By default, Fetch **does not** reject HTTP errors (like 404 or 500 status codes). You'll need to **manually** check the response.ok property.
- Fetch returns a **ReadableStream** as the response body, which needs to be processed (e.g., using response.json() for JSON data).



@developer.mohit

# Browser Compatibility

## Axios

- Axios is a JavaScript library that works **in all modern browsers**, including older versions of Internet Explorer (IE).

## Fetch API

- Fetch is **widely** supported in modern browsers but does not work in Internet Explorer without **polyfills**

*a piece of computer code written to implement a feature in a browser that does not yet support it*



@developer.mohit

## Node.js Support

### Axios

- Axios works in both browser and Node.js environments, making it a more versatile choice for **full-stack development**.

### Fetch API

- Fetch is primarily a browser API, though there are polyfills available to use it in **Node.js**.



@developer.mohit

## Built-in Features

### Axios

- Axios **comes with a** richer feature set, including support for automatic transformations of JSON data, query parameters serialization, request timeouts, and automatic data transformation for POST requests.

### Fetch API

- Fetch is more **minimalistic**, which means you might need to manually implement features like request timeouts, retries, or handling complex data transformations.



@developer.mohit



## Summary

### Axios

- Axios is a better choice if you need more control over requests, better error handling, and want to use it across different environments (browser and Node.js). It also provides a more user-friendly API with built-in features that make working with **HTTP requests easier**.

### Fetch API

- Fetch is ideal if you want a **lightweight solution** with modern syntax and you don't need the extra features provided by Axios.



@developer.mohit



# Thank you



**Like**



**Comment**



**Share**



@developer.mohit