

1. Axios

Installation

```
npm install axios
```

Import

```
import axios from "axios";
```

Syntax

```
axios.get(url, config)
axios.post(url, data, config)
axios.put(url, data, config)
axios.delete(url, config)
```

Example — GET Request

```
axios.get("https://jsonplaceholder.typicode.com/users")
  .then(response => console.log(response.data))
  .catch(error => console.error(error));
```






Example — POST Request

```
axios.post("https://jsonplaceholder.typicode.com/posts", {
  title: "Hello World",
  body: "This is my first post",
  userId: 1,
})
  .then(response => console.log(response.data))
  .catch(error => console.error(error));
```

Configuration Example

```
axios({
  method: "post",
  url: "/api/register",
  data: { username: "Gokul", password: "12345" },
  headers: { "Content-Type": "application/json" },
});
```

Features

-  Automatically transforms JSON data.
-  Supports interceptors (e.g., add tokens or handle errors globally).
-  Supports request cancellation.
-  Automatically parses JSON response.
-  Easier error handling.

Example — Interceptors

```
axios.interceptors.request.use(config => {
  config.headers.Authorization = `Bearer
${localStorage.getItem("token")}`;
  return config;
});
```

2. Fetch API

Built-in (no installation needed)

```
fetch(url, options)
  .then(response => response.json())
  .then(data => console.log(data))
```

```
.catch(error => console.error(error));
```

Example — GET Request

```
fetch("https://jsonplaceholder.typicode.com/users")
  .then(response => response.json()) // must convert to JSON
  manually
  .then(data => console.log(data))
  .catch(error => console.error(error));
```

Example — POST Request

```
fetch("https://jsonplaceholder.typicode.com/posts", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
    title: "Hello World",
    body: "This is my first post",
    userId: 1,
  }),
})
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error(error));
```

Options

```
{
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify(data),
}
```

Features

-  Native in browsers (no extra library).

- ⚠ Requires manual JSON conversion.
- ⚠ Less user-friendly error handling (only rejects on network failure).
- ⚠ No built-in request cancellation (needs AbortController).

3. Key Differences

Feature	Axios	Fetch API
Library	External (install required)	Built-in
JSON Parsing	Automatic	Manual (.json())
Error Handling	Rejects for HTTP & network errors	Rejects only for network errors
Timeout Support	Built-in	Manual using AbortController
Upload/Download Progress	Supported	Needs ReadableStream
Interceptors	Yes	No
Simplicity	More concise	Slightly verbose

4. Example Comparison (POST)

Axios

```
await axios.post("/api/users", { name: "Gokul" });
```

Fetch

```
await fetch("/api/users", {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify({ name: "Gokul" }),
});
```

When to Use What

Use Case	Recommendation
You need interceptors, cancel tokens, or timeouts	Axios
You want zero dependencies	Fetch API
You're using older browsers	Axios (better compatibility)
You're building a simple modern web app	Fetch API is fine

Quick Reference

Axios Example

```
axios.get('/api/data').then(res => console.log(res.data));
```

Fetch Example

```
const res = await fetch('/api/data');  
const data = await res.json();  
console.log(data);
```