**📦 Axios – HTTP Client for JavaScript**

**🔹 What is Axios?**

\*\*Axios\*\* is a popular JavaScript library used to make HTTP requests from the browser (or Node.js). It's built on top of Promises and provides a clean, easy-to-use API for working with APIs or backend services.

**✅ Why Use Axios?**

Promise-based\*\*: Works perfectly with `async/await`.

Simple syntax\*\* for common tasks like `GET`, `POST`, `PUT`, `DELETE`.

Automatic JSON parsing\*\* from server responses.

\*Built-in \*\*error handling\*\*.

Easily set \*\*custom headers\*\* (like authentication tokens).

Works well with \*\*React\*\* and other frontend frameworks.

**🔧 Installing Axios**

Install via npm or yarn:

```bash

npm install axios

# or

yarn add axios

```

Then import it:

```js

import axios from 'axios';

```

---

**🔹 Basic Axios Usage**

**🟢 GET Request**

```js

axios.get('https://api.example.com/users')

.then(response => console.log(response.data))

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

```

**🟠 POST Request**

```js

axios.post('https://api.example.com/users', {

name: 'Alice',

age: 25

})

.then(response => console.log(response.data))

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

```

**🔄 Using Axios with `async/await`**

```js

async function fetchUsers() {

try {

const response = await axios.get('https://api.example.com/users');

console.log(response.data);

} catch (error) {

console.error('Error fetching users:', error.message);

}

}

```

**⚠️ Handling Errors**

Axios allows detailed error handling:

```js

try {

const response = await axios.get('/api/data');

} catch (error) {

if (error.response) {

// Server responded with status code outside 2xx

console.error('Server Error:', error.response.status);

} else if (error.request) {

// Request made but no response received

console.error('No Response:', error.request);

} else {

// Something else happened

console.error('Error:', error.message);

}

}