

# Axios in React

Axios in React simplifies HTTP requests using a promise-based API for methods like `get`, `post`, `put`, `delete`, and `patch`, offering features like **automatic JSON transformation, request/response interception, cancellation, timeout settings, and built-in error handling** (rejecting on 4xx/5xx status codes). Key properties include `response.data` for payload, `error.response` for detailed HTTP errors, `config` for settings (headers, timeouts), and `axios.create()` for custom instances, making API communication clean and efficient in React apps.

This video provides a tutorial on making API requests with Axios in React:

[Dave Gray](#)

[YouTube • 3 Aug 2021](#)

## Key Methods

- `axios(config)`: The base method, takes a config object for any request type.
- `axios.get(url, config)`: Fetches data from a URL.
- `axios.post(url, data, config)`: Sends data to create a resource.
- `axios.put(url, data, config)`: Updates an existing resource.
- `axios.delete(url, config)`: Deletes a resource.
- `axios.patch(url, data, config)`: Partially updates a resource.
- `axios.head(url, config)`: Gets headers without response body.
- `axios.options(url, config)`: Checks server options.

## Core Properties & Features

- **Promises**: Handles async operations with `.then()` (success) and `.catch()` (error).
- **Automatic JSON**: Converts JSON responses to JavaScript objects automatically.
- **Interceptors**: Modify requests or responses globally before handling (e.g., adding auth tokens).
- **Error Handling**: Throws errors for non-2xx status codes (404, 500), unlike Fetch.
- **Timeouts**: Set `timeout` in config to prevent hanging requests.
- **Cancellation**: Cancel ongoing requests using `CancelToken` or `AbortController`.
- `axios.create()`: Creates reusable instances with default configs (base URL, headers).

This video explains how to create custom Axios hooks for your React application:

[Code Radiance](#)

[YouTube • 11 Feb 2024](#)

## Common React Usage Pattern

javascript

```
import axios from 'axios';
import React, { useEffect, useState } from 'react';

function MyComponent() {
  const [data, setData] = useState(null);

  useEffect(() => {
    axios.get('https://api.example.com/items') // .get(url, { params: {} }) for query params
      .then(response => {
        setData(response.data); // Access data here
      })
      .catch(error => {
        console.error('Error fetching data:', error); // Detailed error handling
      });
  }, []);

  return (
    <div>
      {data ? <pre>{JSON.stringify(data, null, 2)}</pre> : <p>Loading...</p>}
    </div>
  );
}
```

---

Revision #3

Created 29 October 2025 02:43:39 by AI API

Updated 9 December 2025 11:50:44 by AI Channel