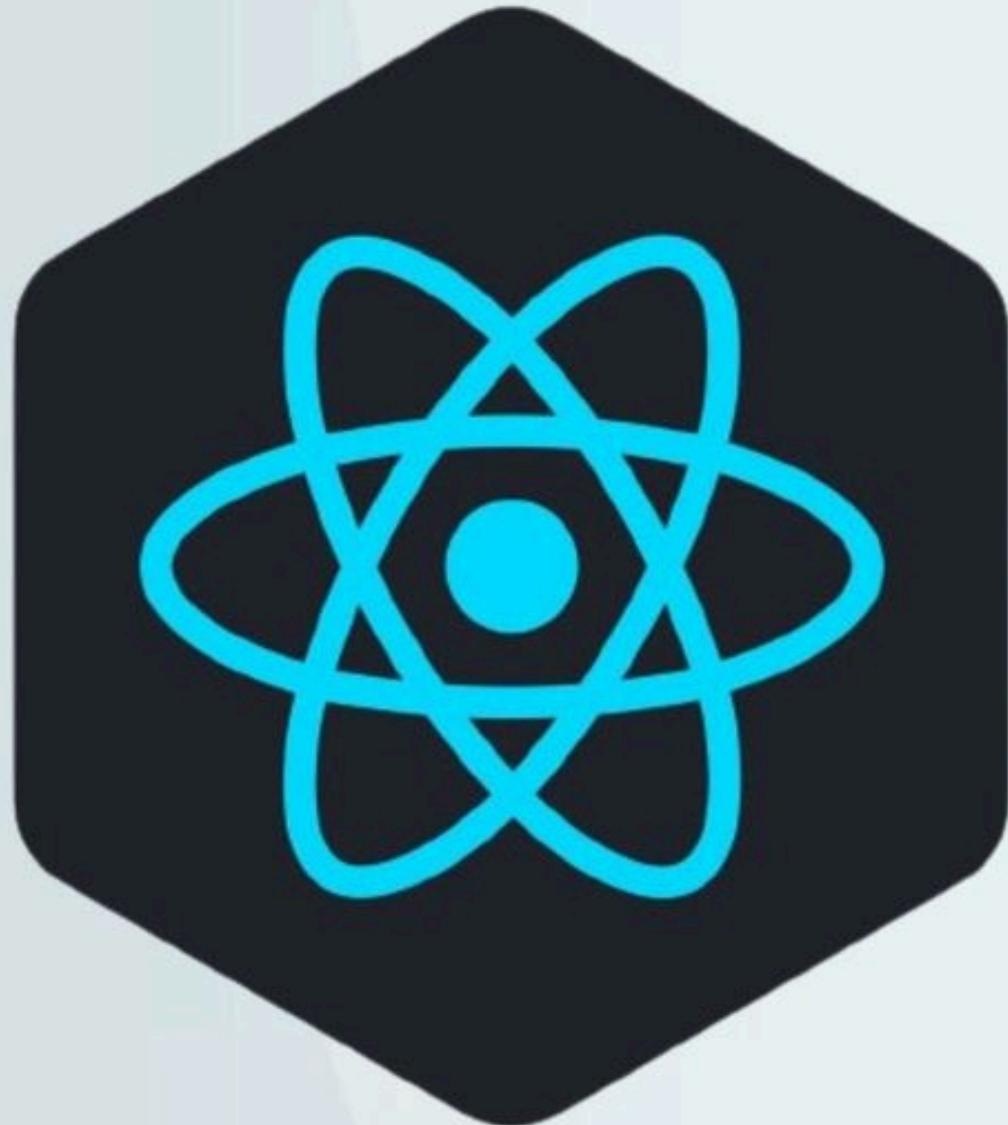


React Axios



What Is Axios In React?

Definition: Axios Is A Promise-Based HTTP Client That Helps In Fetching Data From APIs. In React, It Simplifies The Process Of Making API Requests.

```
import axios from 'axios';

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

- ★ Axios Provides A Cleaner And More Powerful Way To Handle HTTP Requests In React.



Installing Axios In React

How To Install: You Can Easily Install Axios Using NPM Or Yarn.

```
npm install axios // using npm  
yarn add axios // using yarn
```

This Installs Axios Into Your React Project, Ready To Use For API Calls.



Fetching Data Using Axios In React

How To Use: The Best Practice Is To Fetch Data Inside The `useEffect` Hook To Handle Side Effects.

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

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

  useEffect(() => {
    axios.get('https://api.example.com/data')
      .then(response => setData(response.data))
      .catch(error => console.log(error));
  }, []);

  return <div>{JSON.stringify(data)} </div>;
}

export default App;
```

Fetching Data With Axios And `useEffect` Is A Standard Approach In React.



Handling Errors With Axios

Error Handling: It's Important To Handle Errors Gracefully When Fetching Or Posting Data With Axios.

```
● ● ●  
axios.get('https://api.example.com/data')  
  .then(response => console.log(response.data))  
  .catch(error => {  
    if (error.response) {  
      console.log('Error:', error.response.data);  
    }  
  });
```

Proper Error Handling Makes Your React Application More Robust.



Posting Data Using Axios In React

How To Post Data: Axios Also Makes It Easy To Send Data To A Server Using The post Method.

```
function postData() {
  axios.post('https://api.example.com/data', {
    name: 'John',
    age: 30
  })
    .then(response => console.log(response.data))
    .catch(error => console.log(error));
}
```

Axios Helps In Sending Data To The Server Without Much Complexity.



Integrating Axios With Async / Await

Using Async/Await: For Cleaner Code, You Can Use `async/await` Instead Of The Traditional `.then()` Method.

```
async function fetchData() {
  try {
    const response =
      await axios.get('https://api.example.com/data');
    console.log(response.data);
  } catch (error) {
    console.log(error);
  }
}
```

- ★ Async/Await Syntax Makes Your Axios Calls Look Simpler And Easier To Read.





FOLLOW