📝

The Ultimate Axios Guide

In this guide, we are going to see Axios library in React which makes easy to call APIs. So let’s get started.

Table of Content

🆕 What is Axios?

- Many web projects needs to use REST API in their project.
- Axios is lightweight package and use to make HTTP requests in Any Javascript Library like React, Angular or Vue.
- If you use Fetch method in Javascript, Axios is the “Easy to use” Version of Fetch.
- Almost 70% of React users use Axios to make HTTP requests for backend like NodeJS.

☑️ Why we need Axios?

As I said you, Axios is good choice to make Request to Server - Here are some reasons Why we need to use Axios!

- Axios by default Work in JSON format. If you use Fetch, you know we have to first Parse data into JSON format and then we can use it. Axios skip that step itself.

- So we have to use only one .then method in axios.
- Make all types of HTTP requests (GET, POST, PUT, DELETE)
- Better way to Error Handling

🧑‍💻 Use Axios in React

1. First of all, you have to install Axios package in your React Application:

   ```
   npm install axios
   or
   yarn add axios
   ```

   Paste one of this Command in VS Code Terminal and Hit Enter.

2. Import axios from axios package in your Component in Which you want to make HTTP requests. Simple as that.

Example - GET Request with Axios

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

const App = () => {
    const [products, setProducts] = useState([]);

    useEffect(() => {
        //get all products details with GET method
        axios
          .get("https://dummyjson.com/products")
          .then((response) => setProducts(response.data.products));
    }, []);

    return (
        <>
            {products.map((product, index) => {
                return (
                    <h3 key={index}>
                        {product.title} - ${product.price}
                    </h3>
                );
            })}
        </>
    );
```

```
};

export default App;

//Youtube - Code Bless You
```

Example - POST Request with Axios

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

const App = () => {
    const [products, setProducts] = useState([]);

    useEffect(() => {
        //Add new product with POST method
        const product = { title: "Apple 13", price: 799, ... }
        axios
          .post("https://dummyjson.com/products/add", product)
          .then((response) => alert("Product added Successfully!!"););
    }, []);

    return (
        <>
            {products.map((product, index) => {
                return (
                    <h3 key={index}>
                        {product.title} - ${product.price}
                    </h3>
                );
            })}
        </>
    );
};

export default App;

//Youtube - Code Bless You
```

Example - PUT Request with Axios

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

const App = () => {
    const [products, setProducts] = useState([]);

    useEffect(() => {
        //update the product details with PUT method
        const product = { title: "Apple 13", price: 899, ... }
        axios
```

```
          .put("https://dummyjson.com/products/1", product)
          .then((response) => alert("Product Details Updated Successfully!!"););
    }, []);

    return (
        <>
            {products.map((product, index) => {
                return (
                    <h3 key={index}>
                        {product.title} - ${product.price}
                    </h3>
                );
            })}
        </>
    );
};

export default App;

//Youtube - Code Bless You
```

Example - DELETE Request with Axios

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

const App = () => {
    const [products, setProducts] = useState([]);

    useEffect(() => {
        //delete the product-1 with DELETE method
        axios
          .delete("https://dummyjson.com/products/1")
          .then((response) => alert("Product Deleted Successfully!!"););
    }, []);

    return (
        <>
            {products.map((product, index) => {
                return (
                    <h3 key={index}>
                        {product.title} - ${product.price}
                    </h3>
                );
            })}
        </>
    );
};

export default App;

//Youtube - Code Bless You
```

⚠️ Error Handling with Axios

- Let’s see how to handle errors with Axios which is one the easiest thing. Just Add catch method after the then method.
- That will return one parameter error and we can store that error message in any state variable.
- Then you can show that error to users.

Example of Error handling with Axios

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

const App = () => {
    const [products, setProducts] = useState([]);
    const [error, setError] = useState("");

    useEffect(() => {
        axios
          .get("https://dummyjson.com/products")
          .then((response) => setProducts(response.data.products))
          .catch((error) => setError(error.message));
    }, []);

    return (
        <>
            {error !== "" && error}
            {products.map((product, index) => {
                return (
                    <h3 key={index}>
                        {product.title} - ${product.price}
                    </h3>
                );
            })}
        </>
    );
};

export default App;

//Youtube - Code Bless You
```

🚀 My Secret Trick to use Axios

In previous examples, we have to write that API with full URL multiple times which is very tedious task for developers. So we create new resuable variable and then import that variable whenever we need it.

Step 1. Create new file called axios.js

Step 2. Import axios from axios package and then we create new variable called instance. Add baseURL in create function and pass your API baseURL at that place. At last export that instance.

```
import axios from 'axios';

const instance = axios.create({
  baseURL: "https://dummyjson.com"
});

export default instance;
```

Step 3. Now you can use this instance same as axios and you don’t have to pass baseURl every time. (Check this Example)

```
import React, { useEffect, useState } from "react";
import axios from "./axios";

const App = () => {
    const [products, setProducts] = useState([]);
    const [error, setError] = useState("");

    useEffect(() => {
        axios
          .get("/products")
          .then((response) => setProducts(response.data.products))
          .catch((error) => setError(error.message));
    }, []);

    return (
        <>
            {error !== "" && error}
            {products.map((product, index) => {
                return (
                    <h3 key={index}>
                        {product.title} - ${product.price}
                    </h3>
                );
            })}
        </>
    );
};

export default App;
```

```
//Youtube - Code Bless You
```

📌 Axios with Async-await

See our previous code, It looks little confucing. So How can we make it simple and Easy? Solution - Using async-await.

Async - await is the best way to handle Promises in Javascript. So lets create one function (getProducts) with async. With this async we use await in this function.

```
import React, { useEffect, useState } from "react";
import axios from "./axios";

const App = () => {
    const [products, setProducts] = useState([]);
    const [error, setError] = useState("");

    useEffect(() => {
        getProducts();
    }, [])

    async function getProducts() {
        // We can use Await in this function
        const response = await axios.get('/products');
        setProducts(response.data.products);
    }
    return (
        <>
            {error !== "" && error}
            {products.map((product, index) => {
                return (
                    <h3 key={index}>
                        {product.title} - ${product.price}
                    </h3>
                );
            })}
        </>
    );
};

export default App;

//Youtube - Code Bless You
```

Await is like - until we get that it will not run the next line. First of all this first line will run and after we get data in response variable only then this setProducts line will run.

After that we call this function inside useEffect hook. See our code looks Simple and Clean. That’s the power of Async-await.

Always try to use Async-await with axios😉

If you found this guide helpful, share with your classmates and with your friend who need this.

VISIT YOUTUBE CHANNEL : *Code Bless You*