PUBLISHED: JULY 17 2020

React + Axios - HTTP POST Request Examples

Below is a quick set of examples to show how to send HTTP POST requests from React to a backend API using the `axios` HTTP client which is available on npm.

Other HTTP examples available:

- **React + Axios:** GET, PUT, DELETE
- **React + Fetch:** GET, POST, PUT, DELETE
- **Vue + Axios:** GET, POST
- **Vue + Fetch:** GET, POST
- **Angular:** GET, POST, PUT, DELETE
- **Blazor WebAssembly:** GET, POST

SUBSCRIBE FOR UPDATES

Twitter

Facebook

YouTube

GitHub

RSS

- **Axios:** GET, POST, PUT, DELETE
- **Fetch:** GET, POST, PUT, DELETE

Installing axios from npm

With the **npm CLI:** `npm install axios`

With the **yarn CLI:** `yarn add axios`

Simple POST request with a JSON body using axios

This sends an HTTP POST request to the Reqres api which is a fake online REST api that includes a generic `/api/<resource>` route that responds to `POST` requests for any `<resource>` with the contents of the post body and a dynamic id property. This example sends an `article` object to the `/api/articles` route and then assigns the id from the response to the react component state property `articleId` so it can be displayed in the component `render()` method.

```
componentDidMount() {
    // Simple POST request with a JSON body using axios
    const article = { title: 'React POST Request Example'
    axios.post('https://reqres.in/api/articles', article)
        .then(response => this.setState({ articleId: resp
}
```

Example React component at https://stackblitz.com/edit/react-http-post-request-examples-axios?file=App/PostRequest.jsx

POST request using axios with React hooks

This sends the same POST request from React using axios, but this version uses React hooks from a function component instead of lifecycle methods from a traditional React class component. The `useEffect` React hook replaces the `componentDidMount` lifecycle method to make the HTTP POST request when the component loads.

The second parameter to the `useEffect` React hook is an array of dependencies that determines when the hook is run, passing an empty array causes the hook to only be run once when the component first loads, like the `componentDidMount` lifecyle method in a class component. For more info on React hooks see https://reactjs.org/docs/hooks-intro.html.

```
useEffect(() => {
    // POST request using axios inside useEffect React ho
    const article = { title: 'React Hooks POST Request Ex
    axios.post('https://reqres.in/api/articles', article)
        .then(response => setArticleId(response.data.id))

// empty dependency array means this effect will only run
}, []);
```

Example React hooks component at https://stackblitz.com/edit/react-http-post-request-examples-axios?file=App/PostRequestHooks.jsx

POST request using axios with async/await

This sends the same POST request from React using axios, but this version uses an `async` function and the `await` javascript expression to wait for the promises to return (instead of using the promise `then()` method as above).

```
async componentDidMount() {
    // POST request using axios with async/await
    const article = { title: 'React POST Request Example'
    const response = await axios.post('https://reqres.in/
    this.setState({ articleId: response.data.id });
}
```

Example React component at https://stackblitz.com/edit/react-http-post-request-examples-axios?file=App/PostRequestAsyncAwait.jsx

POST request using axios with error handling

This sends a POST request from React using axios to an invalid url on the api then assigns the error message to the `errorMessage` component state property and logs the error to the console.

```
componentDidMount() {
    // POST request using axios with error handling
    const article = { title: 'React POST Request Example'
    axios.post('https://reqres.in/invalid-url', article)
        .then(response => this.setState({ articleId: resp
        .catch(error => {
            this.setState({ errorMessage: error.message }
            console.error('There was an error!', error);
        });
}
```

Example React component at https://stackblitz.com/edit/react-http-post-request-examples-axios?file=App/PostRequestErrorHandling.jsx

POST request using axios with set HTTP headers

This sends the same POST request again from React using axios with a couple of headers set, the HTTP `Authorization` header and a custom header `My-Custom-Header`.

```
componentDidMount() {
    // POST request using axios with set headers
    const article = { title: 'React POST Request Example'
    const headers = {
        'Authorization': 'Bearer my-token',
        'My-Custom-Header': 'foobar'
    };
    axios.post('https://reqres.in/api/articles', article,
        .then(response => this.setState({ articleId: resp
}
```

Example React component at https://stackblitz.com/edit/react-http-post-request-examples-axios?file=App/PostRequestSetHeaders.jsx

Subscribe or Follow Me For Updates

Subscribe to my YouTube channel or follow me on Twitter, Facebook or GitHub to be notified when I post new content.

- Subscribe on YouTube at https://www.youtube.com/JasonWatmore
- Follow me on Twitter at https://twitter.com/jason_watmore

- Follow me on Facebook at https://www.facebook.com/JasonWatmoreBlog
- Follow me on GitHub at https://github.com/cornflourblue
- Feed formats available: RSS, Atom, JSON

Other than coding...

I'm currently attempting to travel around Australia by motorcycle with my wife Tina on a pair of Royal Enfield Himalayans. You can follow our adventures on YouTube, Instagram and Facebook.

- Subscribe on YouTube at https://www.youtube.com/TinaAndJason
- Follow us on Instagram at https://www.instagram.com/tinaandjason
- Follow us on Facebook at https://www.facebook.com/TinaAndJasonVlog

Need Some React Help?

Search fiverr to find help quickly from experienced React developers.

Tags: React, Axios, HTTP, React Hooks

Share: f 🐦

More React Posts

- React - history listen and unlisten with React Router v5
- React Hook Form 7 - Dynamic Form Example with useFieldArray
- React + Fetch - Logout on 401 Unauthorized or 403 Forbidden HTTP Response
- React + Axios - Interceptor to Set Auth Header for API Requests if User Logged In
- React Hook Form - Reset form with default values and clear errors
- React Hook Form - Set form values in useEffect hook after async data load
- React + Fetch - Set Authorization Header for API Requests if User Logged In

- React + Recoil - User Registration and Login Example & Tutorial
- React Hook Form - Password and Confirm Password Match Validation Example
- React Hook Form - Display custom error message returned from API request

Show more...

What do you think?

25 Responses

👍	😝	😍	😯	😤	😥
Upvote	Funny	Love	Surprised	Angry	Sad

0 Comments Jason Watmore's Blog 🔒 Privacy Policy 1 Login

♡ Favorite 3 Tweet f Share Sort by Best

Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.

✉ Subscribe Add Disqus ⚠ Do Not Sell My Data DISQUS

ABOUT

I'm a web developer in Sydney Australia and co-founder of Point Blank Development, I've been building websites and web applications in Sydney since 1998.

Other than coding, I'm currently attempting to travel around Australia by motorcycle with my wife Tina, you can follow our adventure on YouTube and Instagram.

Support me on: Patreon

Find me on:

MONTHS

- 2022
  - January (5)
- 2021
- 2020
- 2019
- 2018
- 2017
- 2016
- 2015
- 2014
- 2013
- 2012
- 2011

TAGS

Alerts, Angular, Angular 10, Angular 11, Angular 2, Angular 4, Angular 5, Angular 6, Angular 7, Angular 8, Angular 9, Angular Directive, Angular UI Router, AngularJS, Animation, ASP.NET, ASP.NET Core, ASP.NET Web API, Authentication and Authorization, AWS, Axios, Azure, Basic Authentication, Blazor, Bootstrap, C#, Chai, CKEditor, CSS3, DDD, Deployment, Design Patterns, Dynamic LINQ, EF Core, ELMAH, ES6, Exchange, Facebook, Fetch, Fluent NHibernate, Formik, Google Analytics, Google API, Google Maps API, Google Plus, Heroku, HTML5, HTTP, IIS, Insecure Content, Instagram API, Ionic Framework, iOS, iPhone, JavaScript, jQuery, JWT, LinkedIn, LINQ, Login, MEAN Stack, MERN Stack, MEVN Stack, Mocha, Modal, MongoDB, Moq, MVC, MVC5, MySQL, .NET, NextJS, NGINX, ngMock, NHibernate, Ninject, NodeJS, npm, Pagination, Pinterest, Razor Pages, React, React Hook Form, React Hooks, Recoil, Redmine, Redux, Registration, Repository, RxJS, Security, Sequelize, Subscribe to Feed: RSS,

Atom, JSON

Code Snippets

Shell Scripting, Sinon, SinonJS, SSH, TDD, Terraform, Twitter, TypeScript, Ubuntu, Umbraco, Unit of Work, Unit Testing, URL Rewrite, Validation, Vanilla JS, VS Code, Vue, Vue 3, Vuex, Webpack, Windows Server 2008

Powered by MEANie | Hosted on DigitalOcean