



Zachary Bennett

How to Send Data Via AJAX In React

Zachary Bennett

Sep 11, 2020 • 4 Min read • 1,679 Views

Sep 11, 2020 • 4 Min read • 1,679 Views

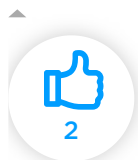
Web Development

Front End Web Dev...

Client-side Framew...

React

Introduction



- [Introduction](#)
- [Using Fetch](#)
- [Using Axios](#)
- [Conclusion](#)
- [Top](#) ^

Introduction

Creating HTTP requests is a fundamental part of any web application. This is no less true when using the React framework. In this guide, you will learn how to send data via AJAX (Asynchronous JavaScript and XML) within your React app. First, you will learn how to compose POST requests by using the browser's built-in Fetch API. After that, you will move on to learning the React best practice of utilizing the HTTP library Axios for composing your requests.

Let's get started!

Using Fetch

When it comes to sending data in your React app, the easiest way to get started is by using the browser's Fetch API. Because the Fetch API is built in to the browser, it is quick and easy to get started. In this section's example, you will see how you can utilize Fetch to create POST requests in order to send data to an external API.

Let's say you have a React application that fetches a list of recipes from an API and then displays them. The best place to make a request for data is within the `componentDidMount` lifecycle method as this will ensure that your component is ready to render data when it is retrieved from the server. The recipe API that we will be using operates off of a simple JSON payload attached to the body of the POST request. The body of a POST request for this API should look like this:

```
1  {  
2    "type": "hot",
```

json

```
3       "limit": 10
4     }
```

This payload will tell the API to retrieve the first 10, hot recipes that are within the database connected to it. You could make this request using Fetch like this:

javascript

```
1   componentDidMount() {
2     const recipeUrl = 'http://localhost:8080/api/recipes';
3     const postBody = {
4       type: "hot",
5       limit: 10
6     };
7     const requestMetadata = {
8       method: 'POST',
9       headers: {
10         'Content-Type': 'application/json'
11       },
12       body: JSON.stringify(postBody)
13     };
14
15     fetch(recipeUrl, requestMetadata)
16       .then(res => res.json())
17       .then(recipes => {
18         this.setState({ recipes });
19       });
20   }
```

In the above example, first we set up some variables which comprise our request. In order to make a POST request using Fetch, you must pass the `fetch` API method a string URL, a request object containing the method (in this case, `POST`), the headers you need to send to the server and the body of the request. As you can see, this method, although quick in terms of getting started, is a little cumbersome. In the next section, you will learn how to use Axios in order to more easily create and send POST requests.

Using Axios

The first thing you will need to do in order to use Axios is to install it. If you are using NPM, you can install Axios easily by running the following command: `npm install --save axios`.

Let's refactor the `componentDidMount` lifecycle method above to use Axios instead of Fetch.

```
1  componentDidMount() {  
2    axios({  
3      method: 'post',  
4      url: 'http://localhost:8080/api/recipes',  
5      data: {  
6        type: "hot",
```

javascript

```
7             limit: 10
8         }
9     })
10     .then(res => this.setState({ recipes: res.data }));
11 }
```

Wow! You can probably see why using Axios to send data via AJAX is considered a best practice within the React ecosystem. Axios makes it much easier to create and send HTTP requests compared to Fetch. On top of this, Axios is backward compatible with older browsers like Internet Explorer 11. So if your application needs to support older browsers, Axios may be a great candidate for you.

Conclusion

In this guide, you learned how to send data via AJAX in your React app. You discovered the fundamentals of composing HTTP POST requests using the browser's built-in Fetch API. You then built on this knowledge by learning how to use Axios, a library built on top of the browser's [XMLHttpRequest](#) API. You learned that Axios is the top choice within the React ecosystem when it comes to selecting an HTTP library and how it can help you easily send data via POST requests.

You can now be confident when it comes to sending data via AJAX within your React app! For more information regarding Axios or Fetch, checkout their respective docs [here](#) and [here](#).

