|  |  |
| --- | --- |
| **ch42.ReactJS Axios Delete Request Example** | **Date: 22-02-2022** |

**Topics**

React Props Validation,

# React Props Validation

In this section, we are going to see the delete request of react axios. We will use the react js to learn the http delete request. Suppose we want to use axios react to send the http delete request. In this case, we have to follow the step-by-step process to send and delete requests in react, which will be described in the given example.

[React](https://www.javatpoint.com/reactjs-tutorial) is a [JavaScript](https://www.javatpoint.com/javascript-tutorial) library. It can build the interface. Sometimes the data need to be revalidated by the reactive JavaScript frameworks such as Angular and React when a page renders. After this, many requests will be lead between various external services and the front end and back end. Using the axios, we can keep things D.R.Y with default, interceptors, and instances. The complex applications use axios, which are frequently making API requests. Different APIs can be created the separate instances. In order to handle the global error, we can set interceptors. For common headers, the default can be unset or set. A little more functionality is provided by axios. The functionality will be useful for those applications that use React.

Axios is a kind of nmp package which is used to send the http request from our application. The api of "jsonplaceholder" uses the axios package to delete the data, and we will use this api in our example. When we use the web to access resources, this process is not instantaneous. The promise API is contained by JavaScript. When we want to perform asynchronous tasks, these promises will be useful. Asynchronous tasks are very useful to perform things in sequence. That means it does not change any things from happening.

**Example:**

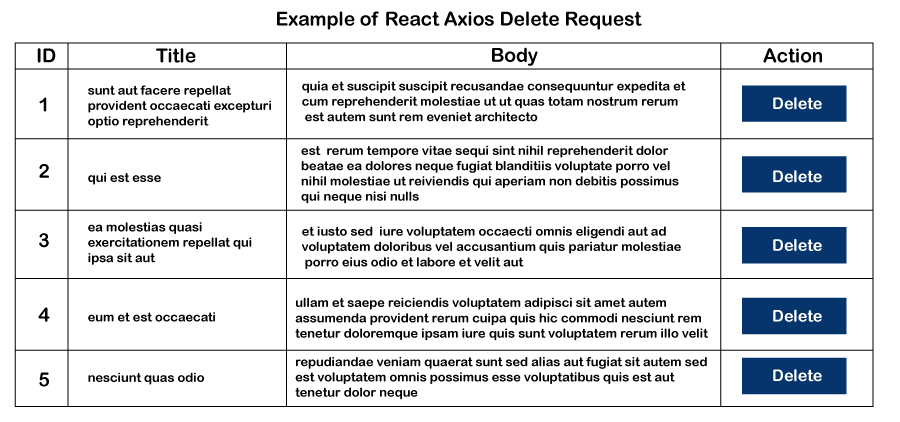
34.7M

739

C++ vs Java

1. **import** React from 'react';
3. **import** axios from 'axios';
5. export **default** **class** PostList **extends** React.Component {
6. state = {
7. posts: []
8. }
10. componentDidMount() {
11. axios.get(`https://jsonplaceholder.typicode.com/posts`)
12. .then(res => {
13. **const** posts = res.data;
14. **this**.setState({ posts });
15. })
16. }
18. deleteRow(id, e){
19. axios.delete(`https://jsonplaceholder.typicode.com/posts/${id}`)
20. .then(res => {
21. console.log(res);
22. console.log(res.data);
24. **const** posts = **this**.state.posts.filter(item => item.id !== id);
25. **this**.setState({ posts });
26. })
28. }
30. render() {
31. **return** (
32. <div>
33. <h1> Example of React Axios Delete Request </h1>
35. <table className="table table-bordered">
36. <thead>
37. <tr>
38. <th>ID</th>
39. <th>Title</th>
40. <th>Body</th>
41. <th>Action</th>
42. </tr>
43. </thead>
45. <tbody>
46. {**this**.state.posts.map((post) => (
47. <tr>
48. <td>{post.id}</td>
49. <td>{post.title}</td>
50. <td>{post.body}</td>
51. <td>
52. <button className="btn btn-danger" onClick={(e) => **this**.deleteRow(post.id, e)}>Delete</button>
53. </td>
54. </tr>
55. ))}
56. </tbody>
58. </table>
59. </div>
60. )
61. }
62. }

After executing the above code, we will see the following output:



Next Topic[React Multiple Checkbox](https://www.javatpoint.com/react-multiple-checkbox)