|  |  |
| --- | --- |
| **ch43.ReactJS Multiple Checkbox** | **Date: 22-02-2022** |

**Topics**

React Props Validation,

# React Props Validation

# React Multiple Checkbox

In this section, we are going to learn the use of multiple checkboxes in React. We will see the use of ReachJS to get all checked values. In our below example, we will show that how we can get value from react multiple checkboxes.

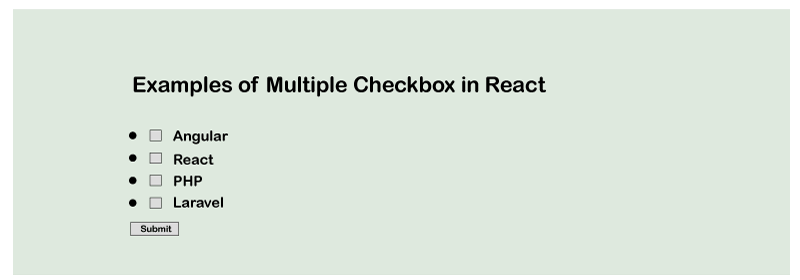
Sometimes we have to set multiple checkboxes on the basis of user requirements. We can set the options to select fruits, and users can select them according to their choices. If users want to select more than one option or multiple options from the list, they can also do this. In this case, we are required to put multiple checkboxes in [ReactJS](https://www.javatpoint.com/reactjs-tutorial). For this, the following example will help us to understand the use of multiple checkboxes in react.

In our below example, we are going to take an array of one category, which will contain React, [Laravel](https://www.javatpoint.com/laravel), [PHP](https://www.javatpoint.com/php-tutorial), Angular, etc. We will show dynamic multiple checkboxes by using the map loop. We will make a variable named "checkedItems" to store the information of the checkbox, which is selected by the user. When the form is submitted by the user, they will get the values of all selected checkboxes. The example code to do this is as follows:

**Example Code:**

1. **import** React, { Component } from 'react';
2. **import** { render } from 'react-dom';
4. **class** App **extends** Component {
5. constructor() {
6. **super**();
7. **this**.state = {
8. categories: [
9. {id: 1, value: "Angular"},
10. {id: 2, value: "React"},
11. {id: 3, value: "PHP"},
12. {id: 4, value: "Laravel"}
13. ],
14. checkedItems: **new** Map()
15. };
17. **this**.handleChange = **this**.handleChange.bind(**this**);
18. **this**.handleSubmit = **this**.handleSubmit.bind(**this**);
19. }
21. handleChange(event) {
22. var isChecked = event.target.checked;
23. var item = event.target.value;
25. **this**.setState(prevState => ({ checkedItems: prevState.checkedItems.set(item, isChecked) }));
26. }
28. handleSubmit(event) {
29. console.log(**this**.state);
30. event.preventDefault();
31. }
33. render() {
34. **return** (
35. <div>
36. <h1> Examples of Multiple Checkbox in React </h1>
38. <form onSubmit={**this**.handleSubmit}>
40. {
41. **this**.state.categories.map(item => (
42. <li>
43. <label>
44. <input
45. type="checkbox"
46. value={item.id}
47. onChange={**this**.handleChange}
48. /> {item.value}
49. </label>
50. </li>
51. ))
52. }
54. <br/>
55. <input type="submit" value="Submit" />
56. </form>
57. </div>
58. );
59. }
60. }
62. render(<App />, document.getElementById('root'));

When we execute the above code, we will get the following output:



Next Topic[React-icons](https://www.javatpoint.com/react-icons)