|  |  |
| --- | --- |
| **ch30.ReactJS Table** | **Date: 22-02-2022** |

**Topics**

React Props Validation,

# React Props Validation

# React Table

A table is an arrangement which organizes information into rows and columns. It is used to store and display data in a structured format.

The react-table is a lightweight, fast, fully customizable (JSX, templates, state, styles, callbacks), and extendable Datagrid built for React. It is fully controllable via optional props and callbacks.

## **Features**

1. It is lightweight at 11kb (and only need 2kb more for styles).
2. It is fully customizable (JSX, templates, state, styles, callbacks).
3. It is fully controllable via optional props and callbacks.
4. It has client-side & Server-side pagination.
5. It has filters.
6. Pivoting & Aggregation
7. Minimal design & easily themeable

## **Installation**

Let us create a **React app** using the following command.

1. $ npx create-react-app myreactapp

Next, we need to install **react-table**. We can install react-table via npm command, which is given below.

1. $ npm install react-table

Once, we have installed react-table, we need to **import** the react-table into the react component. To do this, open the **src/App.js** file and add the following snippet.

1. **import** ReactTable from "react-table";

Let us assume we have data which needs to be rendered using react-table.

1. **const** data = [{
2. name: 'Ayaan',
3. age: 26
4. },{
5. name: 'Ahana',
6. age: 22
7. },{
8. name: 'Peter',
9. age: 40
10. },{
11. name: 'Virat',
12. age: 30
13. },{
14. name: 'Rohit',
15. age: 32
16. },{
17. name: 'Dhoni',
18. age: 37
19. }]

Along with data, we also need to specify the **column info** with **column attributes**.

1. **const** columns = [{
2. Header: 'Name',
3. accessor: 'name'
4. },{
5. Header: 'Age',
6. accessor: 'age'
7. }]

Inside the render method, we need to bind this data with react-table and then returns the react-table.

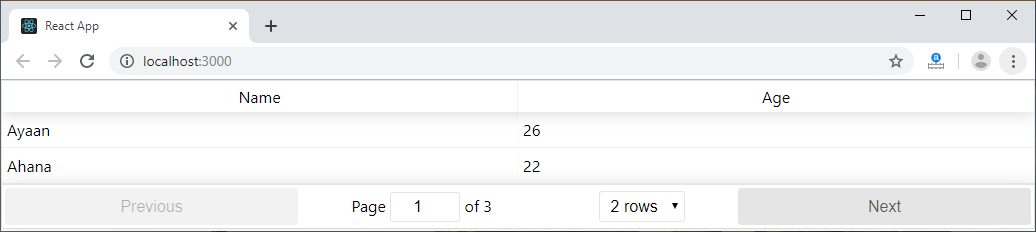
1. **return** (
2. <div>
3. <ReactTable
4. data={data}
5. columns={columns}
6. defaultPageSize = {2}
7. pageSizeOptions = {[2,4, 6]}
8. />
9. </div>
10. )

Now, our **src/App.js** file looks like as below.

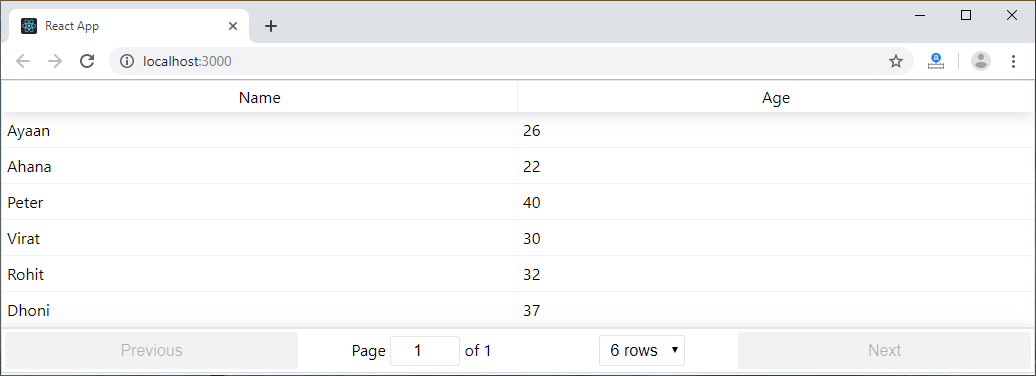
1. **import** React, { Component } from 'react';
2. **import** ReactTable from "react-table";
3. **import** "react-table/react-table.css";
5. **class** App **extends** Component {
6. render() {
7. **const** data = [{
8. name: 'Ayaan',
9. age: 26
10. },{
11. name: 'Ahana',
12. age: 22
13. },{
14. name: 'Peter',
15. age: 40
16. },{
17. name: 'Virat',
18. age: 30
19. },{
20. name: 'Rohit',
21. age: 32
22. },{
23. name: 'Dhoni',
24. age: 37
25. }]
26. **const** columns = [{
27. Header: 'Name',
28. accessor: 'name'
29. },{
30. Header: 'Age',
31. accessor: 'age'
32. }]
33. **return** (
34. <div>
35. <ReactTable
36. data={data}
37. columns={columns}
38. defaultPageSize = {2}
39. pageSizeOptions = {[2,4, 6]}
40. />
41. </div>
42. )
43. }
44. }
45. export **default** App;

**Output**

When we execute the React app, we will get the output as below.



Now, change the rows dropdown menu, we will get the output as below.



Next Topic[Higher-Order Components](https://www.javatpoint.com/react-higher-order-components)