Conclusion

5

Gaurav Singhal

5

Override react-bootstrap with Custom CSS File

Gaurav Singhal

Nov 18, 2020 • 7 Min read • 2,534 Views

Nov 18, 2020 • 7 Min read • 2,534 Views

Web Development Front End Web Dev... Client-side Framew... React

Introduction

The user interface is a crucial part of an app, and impacts the first impression when the end user starts interacting with various pages or functionalities. Generally, the developer uses various UI frameworks with React, such as `material-ui`, `reactstrap`, and `react-bootstrap` to design the multiple components. Still, it may be possible that they need to modify the styles of those components.

`react-bootstrap` contains a set of UI components, and comes with better usability to modify the existing stylesheets. In this guide, you'll learn how to override the `react-bootstrap` component CSS by applying the custom CSS styles to the elements.

Override `react-bootstrap` Table CSS

Table component/elements are one of the primary elements used by most apps where users can see a list of records and edit, paginate, and search over the records. The table element comes with a different variant like simple, striped, expandable, editable, bordered, responsive, etc.

With `react-bootstrap`, the table element can be overridden using the custom CSS classes, but before using the table, you need to import it.

jsx

```jsx
import { Table } from "react-bootstrap";
import "bootstrap/dist/css/bootstrap.min.css";
```

After importing the table element and bootstrap CSS, create the basic table.

Conclusion

jsx

```jsx
render() {
    return (
      <>
        <Table striped bordered hover>
          <thead>
            <tr>
              <th>#</th>
              <th>Name</th>
              <th>Email</th>
              <th>Contact</th>
            </tr>
          </thead>
          <tbody>
            <tr>
              <td>1</td>
              <td>TEST 123</td>
              <td>test@test123.com</td>
              <td>1122334455</td>
            </tr>
            <tr>
              <td>2</td>
              <td>TEST 456</td>
              <td>test@test456.com</td>
              <td>6677889910</td>
            </tr>
            <tr>
              <td>3</td>
              <td>TEST 789</td>
              <td>test@test789.com</td>
              <td>6677889911</td>
            </tr>
          </tbody>
        </Table>
      </>
```

Conclusion

```
        );
    }
```

the above `render()` function, `<Table>` elements uses the header and body section along with the combination of rows and columns.

For example, if you want to change the table border, then create the below CSS class.

css

```css
.table-bordered {
  border: 5px double orange !important;
}
```

`table-bordered` is an official class for the table implemented in the bootstrap, but if you want to override it then you need to define custom CSS with the same name along with the custom values.

Once you run the example, you can see that the table border is changed to orange as per the custom styles defined. Next, you can modify the row hover color.

css

```css
.table-hover tbody tr:hover {
    color: yellow !important;
    background-color: brown;
}
```

Conclusion

Once you apply the above style, you will hover the rows with the updated color combination.

ow is the complete code of the table implementation.

jsx

```jsx
import React, { Component } from "react";
import { Table } from "react-bootstrap";
import "bootstrap/dist/css/bootstrap.min.css";

export class Example1 extends Component {
  render() {
    return (
      <>
        <Table striped bordered hover>
          <thead>
            <tr>
              <th>#</th>
              <th>Name</th>
              <th>Email</th>
              <th>Contact</th>
            </tr>
          </thead>
          <tbody>
            <tr>
              <td>1</td>
              <td>TEST 123</td>
              <td>test@test123.com</td>
              <td>1122334455</td>
            </tr>
            <tr>
              <td>2</td>
              <td>TEST 456</td>
              <td>test@test456.com</td>
              <td>6677889910</td>
            </tr>
            <tr>
              <td>3</td>
              <td>TEST 789</td>
              <td>test@test789.com</td>
              <td>6677889911</td>
            </tr>
          </tbody>
        </Table>
      </>
    );
  }
}

export default Example1;
```

After adding all the custom CSS, you can see that the custom styles get applied, and the previous CSS behavior is entirely different.

Override `react-bootstrap` Form Controls Such as Button

You have seen the example of the `<Table>` element where the existing CSS gets updated with the custom styles, and in the same way, any form of controls can be overridden.

Conclusion

Form controls may contain various input elements such as input, button, dropdown, radio button, checkbox, file upload, etc.

u will see one example of modifying the CSS of the `<Button>` element.

Import the button controls from `react-bootstrap`, as given below.

jsx

```
import { Button } from "react-bootstrap";
import "bootstrap/dist/css/bootstrap.min.css";
```

Use the button component into the `render()` function as a primary variant.

jsx

```
render() {
    return (
      <>
        <Button variant="primary">
          Primary
        </Button>
      </>
    );
}
```

The button color will show primary color because the `variant` props have value as `primary`, but you need to change the color according to the customer requirement.

You can create the custom class and related CSS, as given below.

css

```css
.custom-btn {
  background-color: blueviolet !important;
  border: #fff !important;
}
```

As you can see in the above custom class, the button's background color and its border are defined. But if you want to apply those custom CSS, you can use `className` props as given below.

jsx

```jsx
render() {
    return (
      <>
        <Button variant="primary" className="custom-btn">
          Primary
        </Button>
      </>
    );
}
```

Below is the complete example of additional props added along with the `<Button>` element, which uses the custom CSS styles.

Conclusion

5

5

jsx

```jsx
import React, { Component } from "react";
import { Button } from "react-bootstrap";
import "bootstrap/dist/css/bootstrap.min.css";

export class Example2 extends Component {
  constructor(props) {
    super(props);
  }

  render() {
    return (
      <>
        <Button variant="primary" className="custom-btn">
          Primary
        </Button>
      </>
    );
  }
}

export default Example2;
```

Once you run the above example, you can see that the custom CSS styles replace the existing color combination to override any current styles and classes using `className` props of the element.

Conclusion

5

Conclusion

5

Every UI framework contains components that should be able to be designed or that have capabilities to override existing CSS styles. These skills also apply to `react-bootstrap`, which provides the ability to modify the current styles to some extent.

It's always handy to override CSS because components may need to be modified based on the project requirements.