ReactJS - Styling

In general, React allows component to be styled using CSS class through className attribute. Since, the React JSX supports JavaScript expression, a lot of common CSS methodology can be used. Some of the top options are as follows −

CSS stylesheet − Normal CSS styles along with *className*
Inline styling − CSS styles as JavaScript objects along with camelCase properties.
CSS Modules − Locally scoped CSS styles.
Styled component − Component level styles.
Sass stylesheet − Supports Sass based CSS styles by converting the styles to normal css at build time.
Post processing stylesheet − Supports Post processing styles by converting the styles to normal css at build time.

Let use learn how to apply the three important methodology to style our component in this chapter.

CSS Stylesheet
Inline Styling
CSS Modules

CSS Stylesheet

CSS stylesheet is usual, common and time-tested methodology. Simply create a CSS stylesheet for a component and enter all your styles for that particular component. Then, in the component, use className to refer the styles.

Let us style our *ExpenseEntryItem* component.

Open *expense-manager* application in your favorite editor.

Next, open *ExpenseEntryItem.css* file and add few styles.

```
div.itemStyle {
   color: brown;
   font-size: 14px;
}
```

Next, open *ExpenseEntryItem.js* and add *className* to the main container.

```
import React from 'react';
import './ExpenseEntryItem.css';

class ExpenseEntryItem extends React.Component {
   render() {
      return (
         <div className="itemStyle">
            <div><b>Item: </b> <em>Mango Juice</em></div>
            <div><b>Amount: </b> <em>30.00</em></div>
            <div><b>Spend Date: </b> <em>2020-10-10</em></div>
            <div><b>Category: </b> <em>Food</em></div>
         </div>
      );
   }
}
export default ExpenseEntryItem;
```

Next, serve the application using npm command.

```
npm start
```

Next, open the browser and enter *http://localhost:3000* in the address bar and press enter.

CSS Stylesheet

Image not found or type unknown

CSS stylesheet is easy to understand and use. But, when the project size increases, CSS styles will also increase and ultimately create lot of conflict in the class name. Moreover, loading the CSS file directly is only supported in Webpack bundler and it may not supported in other tools.

Inline Styling

Inline Styling is one of the safest ways to style the React component. It declares all the styles as *JavaScript objects* using DOM based css properties and set it to the component through *style* attributes.

Let us add inline styling in our component.

Open *expense-manager* application in your favorite editor and modify *ExpenseEntryItem.js* file in the src folder. Declare a variable of type object and set the styles.

```
itemStyle = {
   color: 'brown',
   fontSize: '14px'
}
```

Here, *fontSize* represent the css property, font-size. All css properties can be used by representing it in *camelCase* format.

Next, set *itemStyle* style in the component using curly braces {} −

```
render() {
   return (
      <div style={ this.itemStyle }>
         <div><b>Item:</b> <em>Mango Juice</em></div>
         <div><b>Amount:</b> <em>30.00</em></div>
         <div><b>Spend Date:</b> <em>2020-10-10</em></div>
         <div><b>Category:</b> <em>Food</em></div>
      </div>
   );
}
```

Also, style can be directly set inside the component −

```
render() {
   return (
      <div style={
         {
            color: 'brown',
            fontSize: '14px'
         }
      }>
         <div><b>Item:</b> <em>Mango Juice</em></div>
         <div><b>Amount:</b> <em>30.00</em></div>
         <div><b>Spend Date:</b> <em>2020-10-10</em></div>
         <div><b>Category:</b> <em>Food</em></div>
      </div>
   );
}
```

Now, we have successfully used the inline styling in our application.

Next, serve the application using npm command.

```
npm start
```

Next, open the browser and enter *http://localhost:3000* in the address bar and press enter.

Inline Styling

Image not found or type unknown

CSS Modules

Css Modules provides safest as well as easiest way to define the style. It uses normal css stylesheet with normal syntax. While importing the styles, CSS modules converts all the styles into locally scoped styles so that the name conflicts will not happen. Let us change our component to use *CSS modules*

Open expense-manager application in your favorite editor.

Next, create a new stylesheet, ExpenseEntryItem.module.css file under src/components folder and write regular css styles.

```
div.itemStyle {
   color: 'brown';
   font-size: 14px;
}
```

Here, file naming convention is very important. React toolchain will pre-process the css files ending with *.module.css* through *CSS Module*. Otherwise, it will be considered as a normal stylesheet.

Next, open *ExpenseEntryItem.js* file in the *src/component* folder and import the styles.

```
import styles from './ExpenseEntryItem.module.css'
```

Next, use the styles as JavaScript expression in the component.

```
<div className={styles.itemStyle}>
```

Now, we have successfully used the CSS modules in our application.

The final and complete code is −

```
import React from 'react';
import './ExpenseEntryItem.css';
import styles from './ExpenseEntryItem.module.css'

class ExpenseEntryItem extends React.Component {
   render() {
      return (
         <div className={styles.itemStyle} >
            <div><b>Item: </b> <em>Mango Juice</em></div>
            <div><b>Amount: </b> <em>30.00</em></div>
            <div><b>Spend Date: </b> <em>2020-10-10</em></div>
```

```
            <div><b>Category: </b> <em>Food</em></div>
          </div>
        );
    }
}
export default ExpenseEntryItem;
```

Next, serve the application using npm command.

```
npm start
```

Next, open the browser and enter *http://localhost:3000* in the address bar and press enter.

CSS Modules

Image not found or type unknown

Revision #1
Created 14 December 2022 10:40:08 by Admin
Updated 14 December 2022 10:40:55 by Admin