

Styling Components

In this section, we'll add some basic styling to our application and components using the `src/App.css` and `src/index.css` files.

These two files `src/App.css` and `src/index.css` should already be in your project since you have bootstrapped it with *create-react-app*. They should be imported in your `src/App.js` and `src/index.js` files too. The following is CSS that can be copied and pasted to these files but feel free to use your own if you're comfortable with CSS.

First, styling for our overall application:

`src/index.css`

```
body {
  color: #222;
  background: #f4f4f4;
  font: 400 14px CoreSans, Arial, sans-serif;
}

a {
  color: #222;
}

a:hover {
  text-decoration: underline;
}

ul, li {
  list-style: none;
  padding: 0;
  margin: 0;
}

input {
  padding: 10px;
  border-radius: 5px;
  outline: none;
  margin-right: 10px;
  border: 1px solid #dddddd;
}

button {
  padding: 10px;
  border-radius: 5px;
```

```
border: 1px solid #dddddd;
background: transparent;
color: #808080;

cursor: pointer;
}

button:hover {
  color: #222;
}

*:focus {
  outline: none;
}
```

Second, styling for your components in the App file:

`src/App.css`

```
.page {
  margin: 20px;
}

.interactions {
  text-align: center;
}

.table {
  margin: 20px 0;
}

.table-header {
  display: flex;
  line-height: 24px;
  font-size: 16px;
  padding: 0 10px;
  justify-content: space-between;
}

.table-empty {
  margin: 200px;
  text-align: center;
  font-size: 16px;
}

.table-row {
  display: flex;
  line-height: 24px;
  white-space: nowrap;
  margin: 10px 0;
  padding: 10px;
  background: #ffffff;
  border: 1px solid #e3e3e3;
}

.table-header > span {
  overflow: hidden;
  text-overflow: ellipsis;
  padding: 0 5px;
```

```

}

.table-row > span {
  overflow: hidden;
  text-overflow: ellipsis;
  padding: 0 5px;
}

.button-inline {
  border-width: 0;
  background: transparent;
  color: inherit;
  text-align: inherit;
  -webkit-font-smoothing: inherit;
  padding: 0;
  font-size: inherit;
  cursor: pointer;
}

.button-active {
  border-radius: 0;
  border-bottom: 1px solid #38BB6C;
}

```

Now we can use this style with some of our components. Remember to use React `className` instead of `class` as an HTML attribute.

First, apply it in your App ES6 class component:

```

class App extends Component {
  ...

  render() {
    const { searchTerm, list } = this.state;
    return (
      <div className="page">
        <div className="interactions">
          <Search
            value={searchTerm}
            onChange={this.onSearchChange}
          >
            Search
          </Search>
        </div>
        <Table
          list={list}
          pattern={searchTerm}
          onDismiss={this.onDismiss}
        />
      </div>
    );
  }
}

```

Second, apply it in your Table functional stateless component:



```
const Table = ({ list, pattern, onDismiss }) =>
  <div className="table">
    {list.filter(isSearched(pattern)).map(item =>
      <div key={item.objectID} className="table-row">
        <span>
          <a href={item.url}>{item.title}</a>
        </span>
        <span>{item.author}</span>
        <span>{item.num_comments}</span>
        <span>{item.points}</span>
        <span>
          <Button
            onClick={() => onDismiss(item.objectID)}
            className="button-inline"
          >
            Dismiss
          </Button>
        </span>
      </div>
    )}
  </div>
```

Now the application and components have been styled with basic CSS. Further, we know JSX mixes up HTML and JavaScript, and now we could arguably add CSS to that mix. That's called inline style, where you can define JavaScript objects and pass them to the style attribute of an element.

Let's keep the Table column width flexible by using inline style.



```
const Table = ({ list, pattern, onDismiss }) =>
  <div className="table">
    {list.filter(isSearched(pattern)).map(item =>
      <div key={item.objectID} className="table-row">
        <span style={{ width: '40%' }}>
          <a href={item.url}>{item.title}</a>
        </span>
        <span style={{ width: '30%' }}>
          {item.author}
        </span>
        <span style={{ width: '10%' }}>
          {item.num_comments}
        </span>
        <span style={{ width: '10%' }}>
          {item.points}
        </span>
        <span style={{ width: '10%' }}>
          <Button
            onClick={() => onDismiss(item.objectID)}
            className="button-inline"
          >
            Dismiss
          </Button>
        </span>
      </div>
    )}
  </div>
```

```
</span>
</div>
})
</div>
```

The style is inlined now. Define the style objects outside of your elements to make it cleaner.

```
const largeColumn = {
  width: '40%',
};

const midColumn = {
  width: '30%',
};

const smallColumn = {
  width: '10%',
};
```



After that, we use them in the columns: ``. There are different opinions and solutions about styling in React, but the pure inline CSS we used is sufficient for this tutorial.

Further Readings:

I don't want to be opinionated here, but I want to leave you some more options. You can read about them and apply them on your own:

- [styled-components](#)
- [CSS Modules](#)

But if you are new to React, I would recommend sticking to pure CSS and inline style for now.