

Styling With React

Styling your applications with CSS can be tedious. Luckily for us, others have built pre-styled web components in which we can place our react components. There are a number of pre-styled web components out there. It's likely you've already come across [Bootstrap](#), [Material](#), or possibly [Semantic](#). Luckily for us, [React Bootstrap](#) and [React Material Web Components](#), and [Segment](#) already exist for us. By using these pre-defined components, we can stand on the shoulders of giants, and riff off of their styling. While each of these are different in their own right, they're all more or less the same in their structure of how to install them.

Installation:

Each of these have their own installation guides. You just need to make sure that you include the proper links within your header in your `public/index.html` file.

Using:

There's no point in reinventing the wheel. With these packages, you need only to import the required material, and then use them as their own JSX elements. Let's start with adding a linear loader to our page to make it apparent that we're loading data (the remainder of the code is from the previous lesson):

```
1  import React, { useState, useEffect } from "react";
2  import NodeCache from "node-cache";
3  import shuffle from "lodash/shuffle";
4
5  import { LinearProgress } from "@rmwc/linear-progress";
6  import "@material/linear-progress/dist/mdc.linear-progress.css";
7
8  // initializing the cache
9  const myCache = new NodeCache({ stdTTL: 100, checkperiod: 120 });
10
11 export const Reddit = props => {
12   const cacheKey = "subreddits";
13   const cachedStuff = myCache.get(cacheKey);
14   const startingArrays = cachedStuff || [];
15
16   // Hooks
17   const [allPosts, setAllPosts] = useState(startingArrays);
```

```
18  const [searchTerm, setSearchTerm] = useState("");
19  const [posts, setPosts] = useState(startingArrays);
20
21  // create a boolean hook for whether we want to see a loading bar:
22  const [isLoading, setIsLoading] = useState(false);
23
24  const callWithFetch = async () => {
25    if (cachedStuff) {
26      setAllPosts(cachedStuff);
27      setPosts(cachedStuff);
28      return;
29    }
30
31    // Set the is loading bar:
32    setIsLoading(true);
33
34    const urls = [`r/${props.subreddit}.json`, `r/birdswitharms.json`];
35
36    const subredditArrays = urls.map(async url => {
37      const response = await fetch(url);
38      const json = await response.json();
39      const allPosts = json.data.children.map(obj => obj.data);
40
41      return allPosts;
42    });
43
44    const awaitedThings = await Promise.all(subredditArrays);
45    const allPosts = awaitedThings.flat();
46    const shuffledData = shuffle(allPosts);
47
48    myCache.set(cacheKey, shuffledData, 100);
49    setAllPosts(shuffledData);
50    setPosts(shuffledData);
51    setIsLoading(false);
52  };
53
54  useEffect(() => {
55    if (allPosts.length === 0) {
56      callWithFetch();
57    }
58  });
59
60  useEffect(() => {
61    document.title = `Showing ${posts.length} posts`;
62  });
63
```

```

64   const handleSearch = event => {
65     setSearchTerm(event.target.value);
66     setPosts(allPosts.filter(post =>
post.title.includes(event.target.value)));
67   };
68
69
70   const LoadingBarDisplay = () => {
71     return <> {isLoading && <LinearProgress />}</>;
72   };
73
74   return (
75     <div>
76       <h1>r/{props.subreddit}</h1>
77       <div> FILTER THE CUTENESS </div>
78       <input type="text" value={searchTerm} onChange={handleSearch} />
79       <br />
80       <LoadingBarDisplay />
81       <br />
82       <ul>
83         {posts.map(post => (
84           <li key={post.id}>
85             <br />
86             <img src={post.thumbnail} />
87             {post.title}
88           </li>
89         ))}
90       </ul>
91     </div>
92   );
93 };
94

```

Creating New Components:

Now that we have our loading bar set, let's futz with some of the pre-supplied component structure from the documentation. Reddit has a list like structure, but let's revamp it to have cards. First we'll need to take the basic card from the [RMWC documentation](#) and change it up! First we need to make sure that we import the `Card` data and the `Typography` data.

```

1   ...
2
3   import {
4     Card,

```

```

5   CardMedia,
6   CardPrimaryAction,
7   CardActionIcon,
8   CardActions,
9   CardActionIcons,
10  CardMediaContent
11 } from "@rmwc/card";
12 import "@material/card/dist/mdc.card.css";
13 import "@material/button/dist/mdc.button.css";
14 import "@material/icon-button/dist/mdc.icon-button.css";
15
16 import { Typography } from "@rmwc/typography";
17 import "@material/typography/dist/mdc.typography.css";
18
19 ...
20
21 const RedditCard = ({ post }) => {
22   return (
23     <Card style={{ width: "12.5rem" }}>
24       <CardPrimaryAction>
25         <CardMedia
26           square
27           style={{
28             backgroundImage: `url(${post.thumbnail})`
29           }}
30         >
31           <CardMediaContent>
32             <Typography
33               use="subtitle2"
34               tag="div"
35               theme="textPrimaryOnDark"
36               style={{
37                 padding: "0.5rem 1rem",
38                 backgroundImage:
39                   "linear-gradient(to bottom, transparent 0%, rgba(0, 0, 0,
0.5) 100%)",
40                 bottom: "0",
41                 left: "0",
42                 right: "0",
43                 position: "absolute"
44               }}
45             >
46               {post.title}
47             </Typography>
48           </CardMediaContent>
49         </CardMedia>

```

```

50     </CardPrimaryAction>
51     <CardActions>
52       <CardActionIcons>
53         <a href={post.url}>
54           <CardActionIcon icon="comment" />
55         </a>
56         <CardActionIcon onIcon="favorite" icon="favorite_border" />
57         <CardActionIcon icon="bookmark_border" />
58         <CardActionIcon icon="share" />
59       </CardActionIcons>
60     </CardActions>
61   </Card>
62 );
63 };
64
65 ...

```

Now we've got a solid card to display our data!

Adding a Grid for Our Cards:

We have our cards set up, but they're acting as block elements. What we need to do is place them inside of a grid! Luckily for us, instead of having to use flexbox or some other methods of adding grid data, we can just import the [RMWC Layout Grid](#) system:

```

1  ...
2
3  import { Grid, GridCell } from "@rmwc/grid";
4  import "@material/layout-grid/dist/mdc.layout-grid.css";
5
6  ...
7
8
9  return (
10    <div>
11      <h1>r/{props.subreddit}</h1>
12      <div> FILTER THE CUTENESS </div>
13      <input type="text" value={searchTerm} onChange={handleSearch} />
14      <br />
15      <LoadingBarDisplay />
16      <br />
17      <Grid>
18        {posts.map(post => (
19          <GridCell span={3} key={post.id}>
20            <RedditCard post={post} />
21          </GridCell>

```

```

22     }}
23     </Grid>
24 </div>
25 );
26 };

```

Creating a new Component:

We've added our RedditCard to our `APICall.jsx`, but the file is getting too big! So, let's create a brand new stateless component just for our card:

```

1  import React from "react";
2  import {
3    Card,
4    CardPrimaryAction,
5    CardMedia,
6    CardActionIcon,
7    CardActionIcons,
8    CardActionButton,
9    CardActionButtons,
10   CardActions
11 } from "@rmwc/card";
12 import "@material/card/dist/mdc.card.css";
13 import "@material/button/dist/mdc.button.css";
14 import "@material/icon-button/dist/mdc.icon-button.css";
15
16 import { Typography } from "@rmwc/typography";
17 import "@material/typography/dist/mdc.typography.css";
18
19 export const RedditCard = ({ post }) => {
20   console.log(post);
21
22   return (
23     <Card style={{ width: "21rem" }}>
24       <CardPrimaryAction>
25         <CardMedia
26           sixteenByNine
27           style={{
28             backgroundImage: `url(${post.thumbnail})`
29           }}
30         />
31       <div style={{ padding: "0 1rem 1rem 1rem" }}>
32         <Typography use="headline6" tag="h2">
33           {post.title}

```

```

34     </Typography>
35     <Typography
36       use="subtitle2"
37       tag="h3"
38       theme="textSecondaryOnBackground"
39       style={{ marginTop: "-1rem" }}
40     >
41       by {post.author}
42     </Typography>
43   </div>
44 </CardPrimaryAction>
45 <CardActions>
46   <CardActionButtons>
47     <a href={post.url}>
48       <CardActionButton>Read</CardActionButton>
49     </a>
50     <CardActionButton>Bookmark</CardActionButton>
51   </CardActionButtons>
52   <CardActionIcons>
53     <CardActionIcon onIcon="favorite" icon="favorite_border" />
54     <CardActionIcon icon="share" />
55     <CardActionIcon icon="more_vert" />
56   </CardActionIcons>
57 </CardActions>
58 </Card>
59 );
60 };
61

```

Now we've got our APICall.jsx a bit cleaned up:

```

1  import React, { useState, useEffect } from "react";
2  import NodeCache from "node-cache";
3  import shuffle from "lodash/shuffle";
4  import { RedditCard } from "../RedditCard";
5
6  import { LinearProgress } from "@rmwc/linear-progress";
7  import "@material/linear-progress/dist/mdc.linear-progress.css";
8
9  import { Grid, GridCell } from "@rmwc/grid";
10 import "@material/layout-grid/dist/mdc.layout-grid.css";
11
12 // initializing the cache
13 const myCache = new NodeCache({ stdTTL: 100, checkperiod: 120 });
14
15 export const Reddit = props => {

```

```

16  const cacheKey = "subreddits";
17  const cachedStuff = myCache.get(cacheKey);
18  const startingArrays = cachedStuff || [];
19
20  // Hooks
21  const [allPosts, setAllPosts] = useState(startingArrays);
22  const [searchTerm, setSearchTerm] = useState("");
23  const [posts, setPosts] = useState(startingArrays);
24
25  // create a boolean hook for whether we want to see a loading bar:
26  const [isLoading, setIsLoading] = useState(false);
27
28  const callWithFetch = async () => {
29    if (cachedStuff) {
30      setAllPosts(cachedStuff);
31      setPosts(cachedStuff);
32      return;
33    }
34
35    // Set the is loading bar:
36    setIsLoading(true);
37
38    const urls = [`r/${props.subreddit}.json`, `r/birdswitharms.json`];
39
40    const subredditArrays = urls.map(async url => {
41      const response = await fetch(url);
42      const json = await response.json();
43      const allPosts = json.data.children.map(obj => obj.data);
44
45      return allPosts;
46    });
47
48    const awaitedThings = await Promise.all(subredditArrays);
49    const allPosts = awaitedThings.flat();
50    const shuffledData = shuffle(allPosts);
51
52    myCache.set(cacheKey, shuffledData, 100);
53    setAllPosts(shuffledData);
54    setPosts(shuffledData);
55    setIsLoading(false);
56  };
57
58  useEffect(() => {
59    if (allPosts.length === 0) {
60      callWithFetch();
61    }

```



```

62     });
63
64     useEffect(() => {
65         document.title = `Showing ${posts.length} posts`;
66     });
67
68     const handleSearch = event => {
69         setSearchTerm(event.target.value);
70         setPosts(allPosts.filter(post =>
post.title.includes(event.target.value)));
71     };
72
73     const LoadingBarDisplay = () => {
74         return <> {isLoading && <LinearProgress />}</>;
75     };
76
77     return (
78         <div>
79             <h1>r/{props.subreddit}</h1>
80             <div> FILTER THE CUTENESS </div>
81             <input type="text" value={searchTerm} onChange={handleSearch} />
82             <br />
83             <LoadingBarDisplay />
84             <br />
85             <Grid>
86                 {posts.map(post => (
87                     <GridCell span={3} key={post.id}>
88                         <RedditCard post={post} />
89                     </GridCell>
90                 ))}
91             </Grid>
92         </div>
93     );
94 };
95

```