

Another React Exercise: Emoji Cards

Your task is to take a list of emoji objects in JSON format and write a React App to present the data as shown in this picture:



A starting project is provided as a git repository

<https://github.com/beaumoaj/emojiCards.git>. Fork that repository and clone your copy.

Cloning your copy of the repository will create a new sub-folder called **emoji**. Open a terminal window in Ubuntu, **cd** into the new **emoji** folder and run the following commands:

```
npm install
npm start
```

In the **src** sub-folder you will find the files **App.js** and **App.css**. The CSS file provides CSS classes for your App. These include a **row** class that provides a **flex** layout. There are also classes for **card**, **card-emoji**, and **card-keyword**. Finally there is a class called **emoji** to use for the individual emoji characters.

The file called **data.js** contains a list of the emojis that you see in the picture above (and many more besides).

The App component is your starting point.

1 An Emoji component

First you should write a new component called `Emoji` that will display an emoji symbol in a `div`. Follow the steps below to write this component in a file called `Emoji.js` and export it with the name `Emoji`.

I have provided a sample emoji object inside `App.js`. It looks like this:

```
1 const sampleEmoji = {  
2   "title": "Grinning",  
3   "symbol": "😄",  
4   "keywords": "grinning_face_happy_smiley_emotion_emotion"  
5 };
```

The value of the `symbol` property is a special character representing an emoji. If you view it in an editor you may not see anything sensible, but in a web browser this character looks like this smiley face:



1. The `Emoji` component will require a property to be passed to it, and the value of that property is the emoji symbol to display (you could call the property `symbol`).
2. Your `Emoji` component should return a `div` with the `className` of `emoji`. Inside the `div` should be the emoji symbol from the component's properties and nothing else.
3. Once you have written the `Emoji` component, call it from the `App` component. Put the `Emoji` component inside the `<div className="row">` element in the space provided:

```
1 function App() {  
2   return (  
3     <div className="App">  
4       <h1>Emojis</h1>  
5       <div className="row">  
6  
7         </div>  
8     </div>  
9   );  
10 }
```

After completing this step, the output in the browser should now look like this:

Emojis



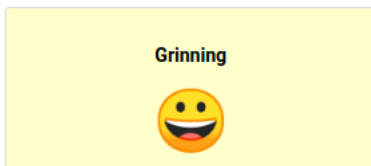
2 An EmojiCard component

The next step is to write a component called `EmojiCard` that will display a card containing a title and an emoji (we will ignore the keywords for now). Follow the steps below to write this component in a file called `EmojiCard.js` and export it with the name `EmojiCard`.

1. Your `EmojiCard` component should have one property called `emoji` whose value will be an emoji object like the one shown above. It will have a `title`, a `symbol`, and some `keywords`.
2. Your `EmojiCard` component should return:
 - a `div` with the `className` set to `"card_card-emoji"`.
 - Inside the `div` you should put a `h3` element which contains the `title` from the emoji object.
 - Immediately below the title you should put your `Emoji` component and pass it the `symbol` to be displayed as its property.
3. Inside your `App` component, delete the `Emoji` component and replace it with your `EmojiCard` component, passing it the `sampleEmoji` as its `emoji` property value.

After completing the above steps, your browser should now display a card containing an emoji with a title as shown below:

Emojis



3 A component for the keywords

the next step is to write a component to display the `keywords` that go with an emoji. These will be included in the emoji card. Follow the steps below to write this component in a file called `Keywords.js` and export it with the name `Keywords`.

1. Your `Keywords` component should have one property called `keywords` whose value will be a string containing a number of keywords.
2. Your component should return a `div` with a `className` set to `row`.
3. Inside that `div` you should use the `split` function to split the list of keywords into an array and then use the `map` function so that, for each `keyword` in the array, you return a `div` with its `className` set to `"card_card-keyword"` and inside that `div` is the `keyword`.

4. Add your `Keywords` component to the end of your `EmojiCard` component, immediately below the `Emoji` component.

After following the steps above, your first emoji card should be displayed in your browser like this:



4 Display all of the emojis

Finally, we are ready to display all of the emojis in the file `data.js`. Note that your `App.js` already contains the line:

```
1 import emojis from "../data.js"
```

The variable `emojis` will contain a list of many emoji objects.

In your `App` component, replace the single call to your `EmojiCard` component with some code to use the `map` function to display each emoji in the list.

After doing that, your browser should now display many emoji cards as shown in the picture at the start of this document.