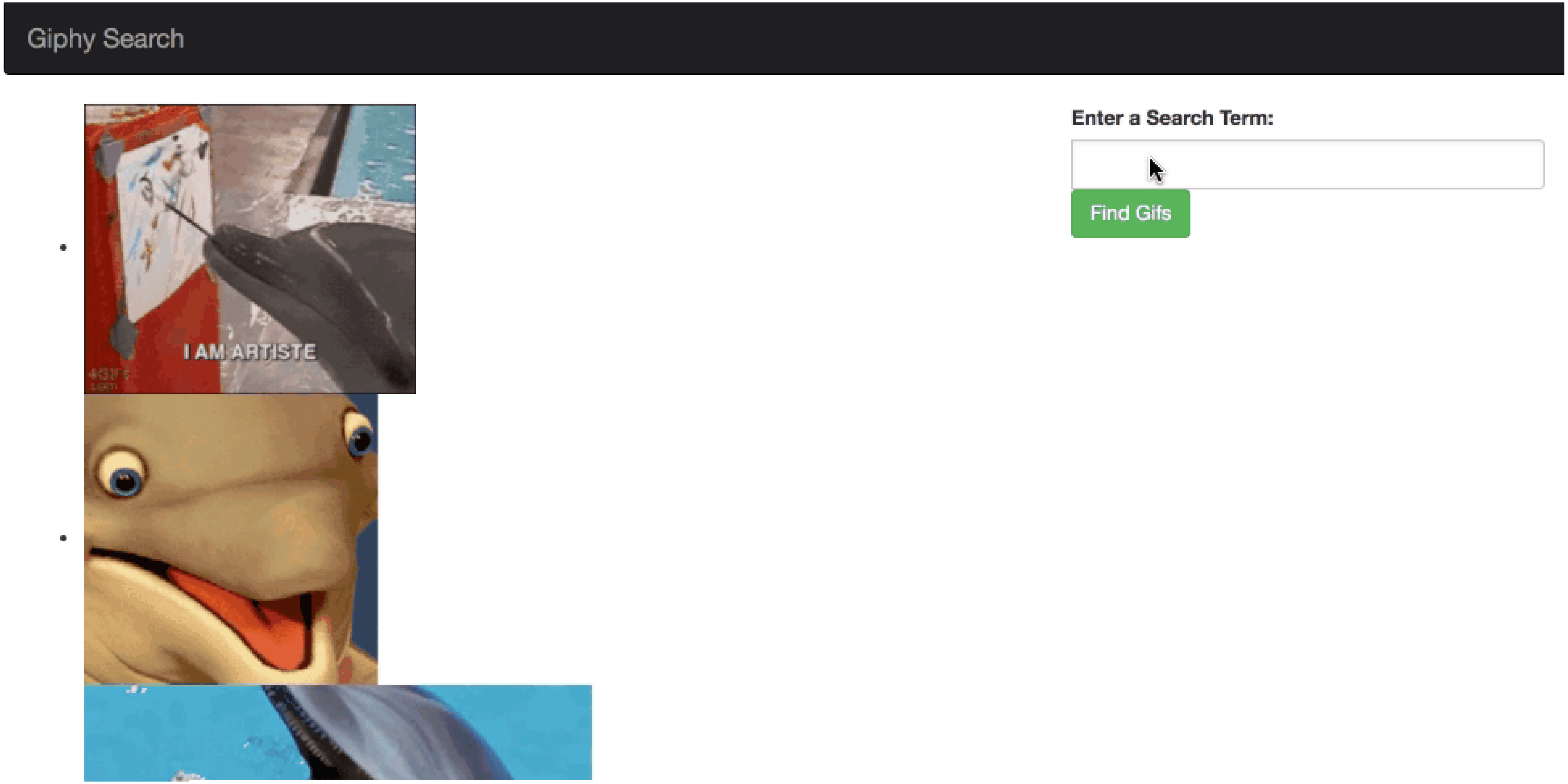
Async Giphy Search Lab

(https://github.com/learn-co-curriculum/react-hooks-async-gif-search-lab) (https://github.com/learn-co-curriculum/react-hooks-async-gif-search-lab/issues/new)

Instructions

You're going to be building out a Gif search using the Giphy API. In this lab, there are no tests to pass. Rather, your task is to create a working app in your browser using the instructions below. When finished, you should have an application that can take in a user's input, fetch JSON data from the Giphy API, and display the results.



Getting Started

The URL for the API is

`https://api.giphy.com/v1/gifs/search?q=YOUR QUERY HERE&api_key=dc6zaTOxFJmzC&rating=g`

While the above API key *may* work, we recommend creating your own API key by following the [instructions on Giphy's developer site (https://developers.giphy.com/docs/api/#quick-start-guide)](https://developers.giphy.com/docs/api/#quick-start-guide) . Creating a key is free and only requires an account. Using your own key will prevent any potential rate limiting if other students are also working on this lesson.

Once you've got your key, you should be able to access the Giphy API from a browser and receive a JSON response to confirm everything is working.

`https://api.giphy.com/v1/gifs/search?q=dolphin&api_key=YOUR API KEY&rating=g`

You should get back an array of objects, each containing information about a particular image.

```
"data": [
  {
    "type": "gif",
    "id": "l0HlNQ03J5JxX6lva",
    "slug": "bbc-wildlife-l0HlNQ03J5JxX6lva",
    "url": "https://giphy.com/gifs/bbc-wildlife-l0HlNQ03J5JxX6lva",
    "bitly_gif_url": "https://gph.is/2iC32M8",
    "bitly_url": "https://gph.is/2iC32M8",

    ...

    "images": {
      "fixed_height_still": {
        "url": "https://media0.giphy.com/media/l0HlNQ03J5JxX6lva/200_s.gif?cid=e1bb72ff5b9fa2866168584b51f13892",
        "width": "400",
        "height": "200",
        "size": "55556"
      },
      ...
      "original": {
        "url": "https://media0.giphy.com/media/l0HlNQ03J5JxX6lva/giphy.gif?cid=e1bb72ff5b9fa2866168584b51f13892",
        "width": "480",
        "height": "240",
      }
      ...
    }
  }
  ]
```

Note: Notice there are many URL keys on each image object. The first `url` key, just below `type` , `id` , and `slug` , will bring you to the images page on **giphy.com** (https://giphy.com/) . We only want the path to the actual image, which is found at `images.original.url` . Using other `url` keys may cause CORS issues.

Your Components

App

Your top level component will be the `App` component - no surprises there! It will be responsible for rendering the `NavBar` component (this component is already provided for you, note the project has bootstrap loaded in) and the `GifListContainer` component.

GifListContainer

`GifListContainer` should be a component that does data fetching and then renders its corresponding sub-component. That’s it.

In our app, the `GifListContainer` will be responsible for fetching the data from the Giphy API, storing the first 3 gifs from the response in its component **state**, and passing that data down to its child, the `GifList` component, as a prop.

It will also render a `GifSearch` component that renders the form. `GifListContainer` should pass down a submit handler function to `GifSearch` as a prop.

GifList

`GifList` receives data from its props and renders html given the input data. It can render a top level `<ul>` with each gif as an `<li>` .

GifSearch

The `GifSearch` component will render a form that receives the user input for the Giphy search. The text input should be a *controlled component* that stores the value of the input in its component state and renders the DOM accordingly. The React component is always in charge of what the DOM looks like.

`GifSearch` should receive a callback prop from its parent. On a submit event, it should invoke that callback prop with the value of the text input. It is this callback function, defined in `GifListContainer` , that will actually query the API with the text the user has entered.