# Building a Simple Dynamic Search Bar in React.js

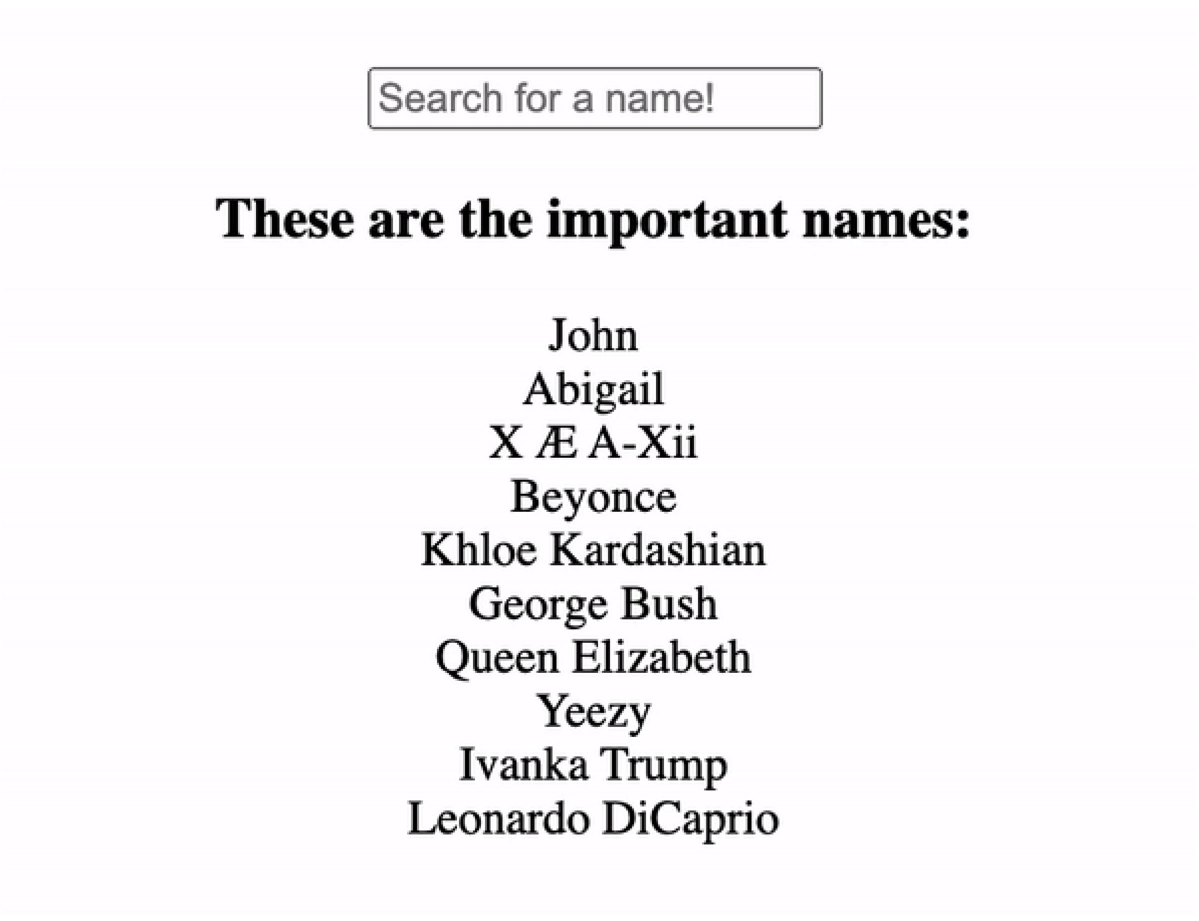
Jonathan Brierre Follow
May 31, 2020 · 4 min read ★

Today, we're going to create a dynamic search bar in React.js!

## What is a dynamic search?

Dynamic searching is the functionality that allows a user to type into a search field to have the results filter and render instantly on the screen.

Here is a gif that shows the functionality in action:



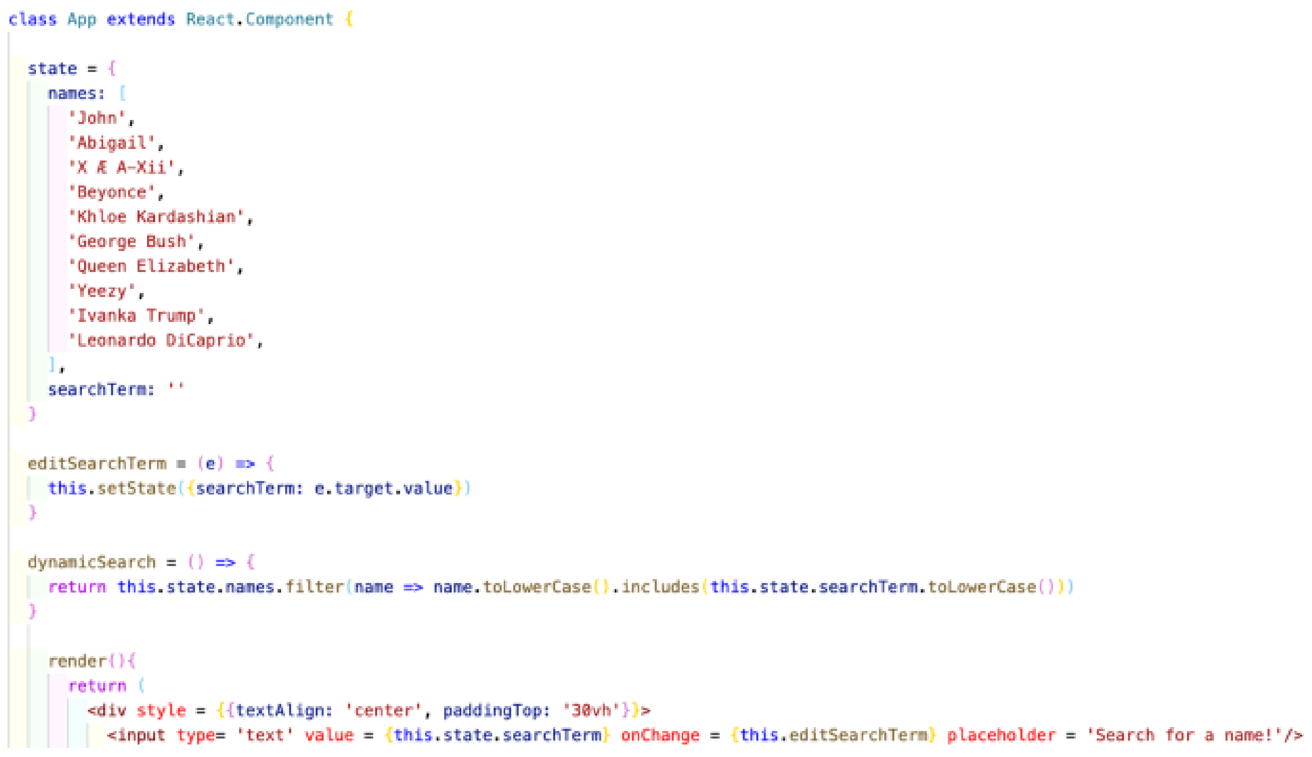
Search for a name!

**These are the important names:**

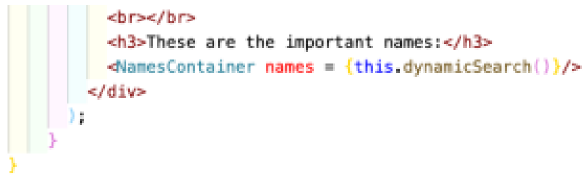
John
Abigail
X Æ A-Xii
Beyonce
Khloe Kardashian
George Bush
Queen Elizabeth
Yeezy
Ivanka Trump
Leonardo DiCaprio

How do we implement such a feature? Let’s keep reading to find out!
By the way, you can find the code referred to in this blog in this public GitHub repo!

Here is a screenshot of my main component, App.js:



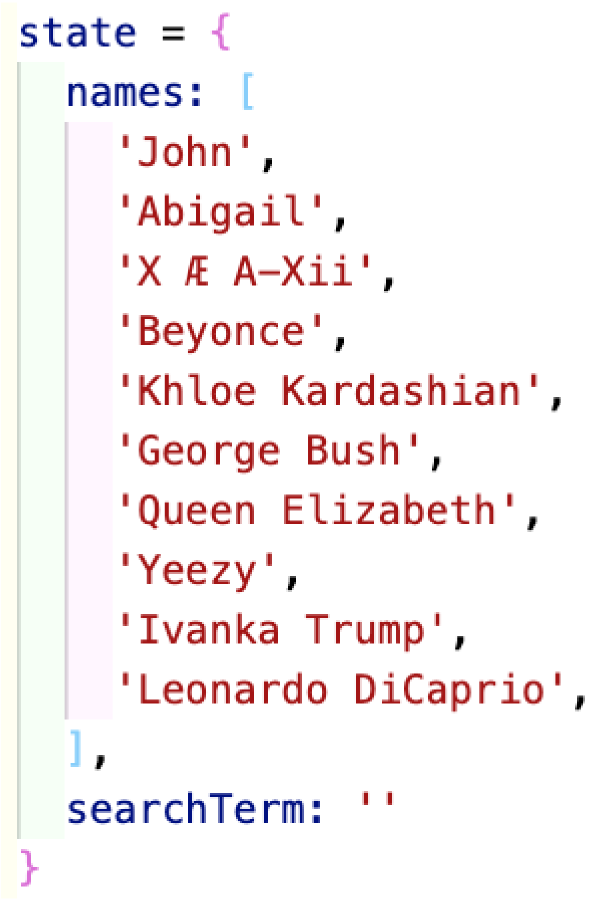
```
class App extends React.Component {

  state = {
    names: [
      'John',
      'Abigail',
      'X Æ A-Xii',
      'Beyonce',
      'Khloe Kardashian',
      'George Bush',
      'Queen Elizabeth',
      'Yeezy',
      'Ivanka Trump',
      'Leonardo DiCaprio',
    ],
    searchTerm: ''
  }

  editSearchTerm = (e) => {
    this.setState({searchTerm: e.target.value})
  }

  dynamicSearch = () => {
    return this.state.names.filter(name => name.toLowerCase().includes(this.state.searchTerm.toLowerCase()))
  }

    render(){
      return (
        <div style = {{textAlign: 'center', paddingTop: '30vh'}}>
          <input type= 'text' value = {this.state.searchTerm} onChange = {this.editSearchTerm} placeholder = 'Search for a name!'/>
```



```
        <br></br>
        <h3>These are the important names:</h3>
        <NamesContainer names = {this.dynamicSearch()}/>
      </div>
    );
  }
}
```

Let's break it down!

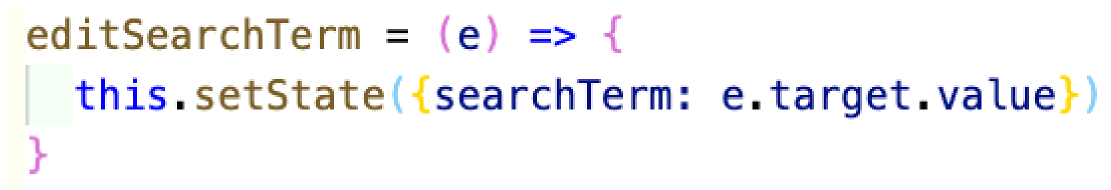
**State:**



```
state = {
  names: [
    'John',
    'Abigail',
    'X Æ A-Xii',
    'Beyonce',
    'Khloe Kardashian',
    'George Bush',
    'Queen Elizabeth',
    'Yeezy',
    'Ivanka Trump',
    'Leonardo DiCaprio',
  ],
  searchTerm: ''
}
```

Our state here contains the key of 'names' pointing to an array of names, and the key of 'searchTerm' referring to an empty string to be controlled by our input JSX tag. If you are unfamiliar with controlling your input fields, I would recommend reading my blog post, '<u>Building a Simple Controlled Form in React.js.</u>'

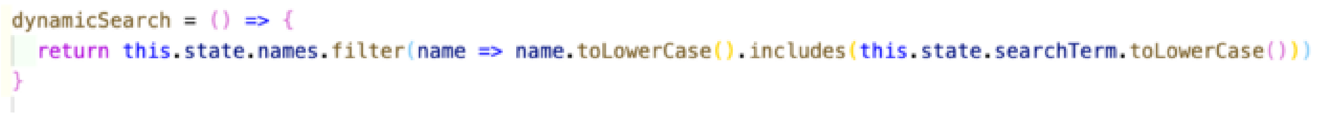
**editSearchTerm Function:**



```
editSearchTerm = (e) => {
  this.setState({searchTerm: e.target.value})
}
```

This function is what allows our input field to control the 'searchTerm' value in our state.

**dynamicSearch function:**



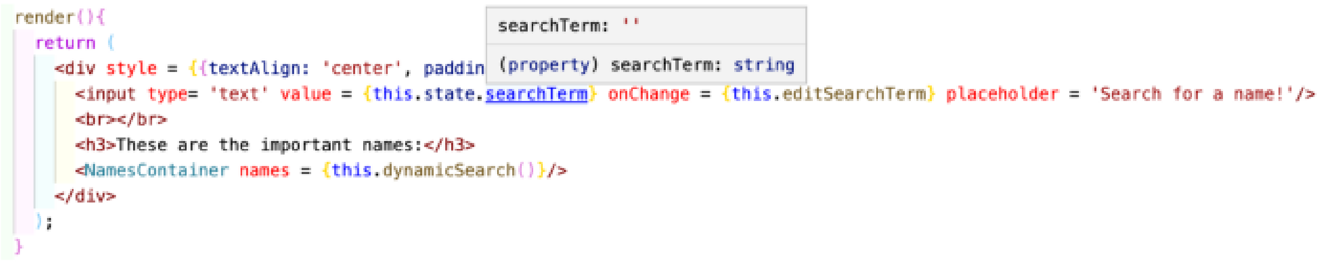
```
dynamicSearch = () => {
  return this.state.names.filter(name => name.toLowerCase().includes(this.state.searchTerm.toLowerCase()))
}
```

This function is what allows us to render what shows up onto the page dynamically. As we type into the input field and control our 'searchTerm' string, we check to see if any of the names have the string within them. If they do, then it would be returned by the filter function. If you are unfamiliar with JavaScript's .filter() higher-order-function, I will refer you to the documentation here.

Keep in mind, when there is nothing in the input field, the dynamicSearch filter function will return the full list of names. Every string will always contain an empty string, the default value of a blank text field.

Also, you may notice that I included the .toLowerCase() function to both the name and the search term. I did this so that when a user types, the input does not need to be case sensitive. If a user were to type in all-caps, the results would filter based on the lowercased values.

**JSX:**

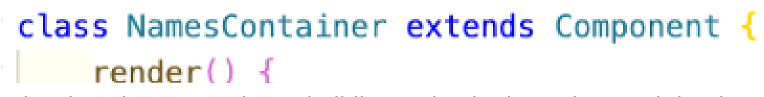


```
render(){
  return (
    <div style = {{textAlign: 'center', paddin
      <input type= 'text' value = {this.state.searchTerm} onChange = {this.editSearchTerm} placeholder = 'Search for a name!'/>
      <br></br>
      <h3>These are the important names:</h3>
      <NamesContainer names = {this.dynamicSearch()}/>
    </div>
  );
}
```

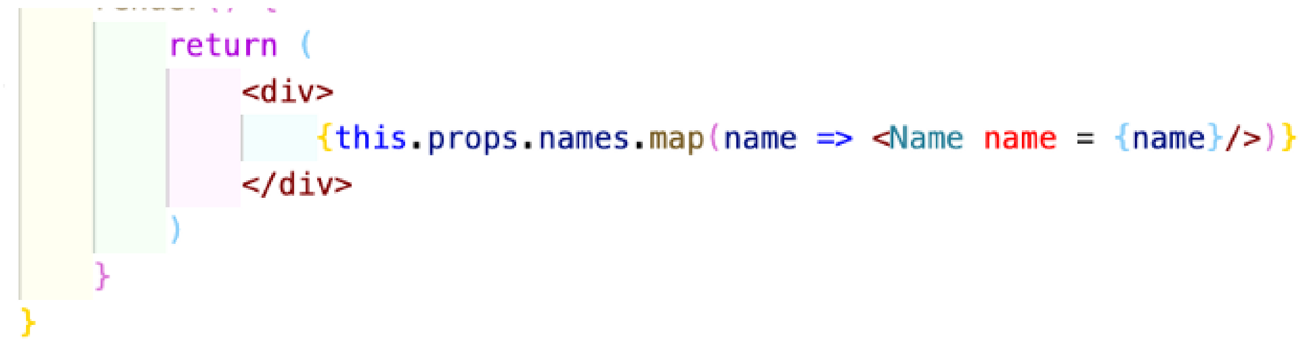
```
searchTerm: ''
(property) searchTerm: string
```

Here you can see our controlled input field as well as a container component for our names called 'NamesContainer.'

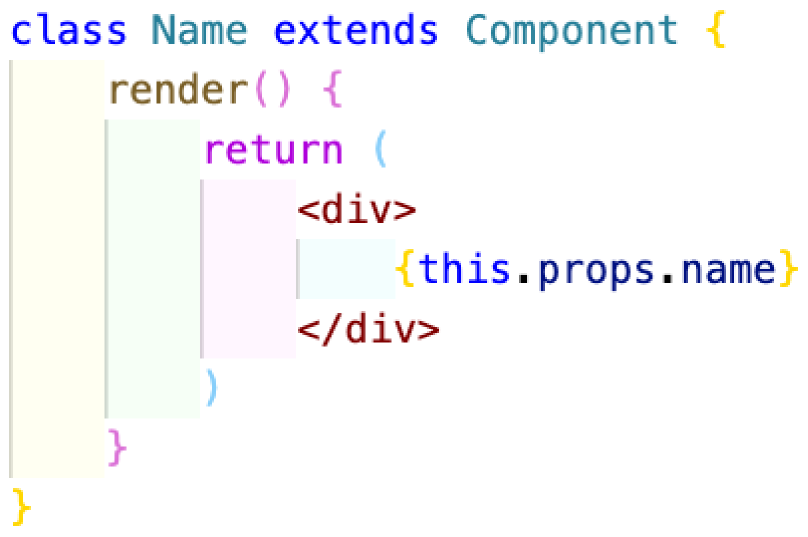
The NamesContainer component takes in an array of names to be parsed into individual name components like so:



```
class NamesContainer extends Component {
    render() {
```

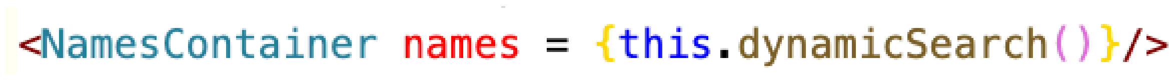


```
        return (
            <div>
                {this.props.names.map(name => <Name name = {name}/>)}
            </div>
        )
    }
}
```



```
class Name extends Component {
    render() {
        return (
            <div>
                {this.props.name}
            </div>
        )
    }
}
```

The key to getting your dynamic search working now is to pass down as props the return value of our dynamicSearch function, the array continuously filtered by our input field. **Note**: We **MUST** invoke the function as we pass it down; otherwise, we wouldn't get the return value, and we'd merely be passing down a function.



```
<NamesContainer names = {this.dynamicSearch()}/>
```

And there you have it — you can now implement this highly-versatile feature into your applications like the pro-coder you aspire to be!

## Resources:

Building a Simple Controlled Form in React.js

Today, we are going to build out a simple controlled form in React.js.

levelup.gitconnected.com

Array.prototype.filter()

The filter() method creates a new array with all elements that pass the test implemented by the provided function...

developer.mozilla.org

jonathanbrierre/Simple-Dynamic-Search

This project was bootstrapped with Create React App. In the project directory, you can run: Runs the app in the...

github.com

JavaScript React Programming Software Development Reactjs