Video 3.7

React App Development

Chris Murphy

Review

- React allows us to create web applications by developing reusable, modular components
- So far, we've seen how to define components within the HTML pages
- How do we develop larger applications with multiple components?

Node.js - Introduction

- **Node.js** is a free, open source platform and framework built in JavaScript

- Includes suite of tools that allows user to prepare JavaScript (and thus React) applications for deployment

- Utilizes Node.js Package Manager (**npm**) to install programs and manage dependencies

Node.js - Benefits

- Instead of including all JavaScript code in a **`<script>`** tag, now we can separate the components into different files to make code more modular

- Node.js allows us to incorporate dependencies of the code within the current file

```
var React = require('react');
var ReactDOM = require('react-dom');

import MyComponent from './MyComponent.js';
```

Node.js - Benefits

- Instead of including all JavaScript code in a **<script>** tag, now we can separate the components into different files to make code more modular

- Node.js allows us to incorporate dependencies of the code within the current file

```
var React = require('react');
var ReactDOM = require('react-dom');

import MyComponent from './MyComponent.js';
```

Node.js - Benefits

- Instead of including all JavaScript code in a **<script>** tag, now we can separate the components into different files to make code more modular

- Node.js allows us to incorporate dependencies of the code within the current file

```
var React = require('react');
var ReactDOM = require('react-dom');

import MyComponent from './MyComponent.js';
```

Node.js - Installation

- Navigate to https://nodejs.org/en/download/ and download and install Node.js and appropriate packages

- Although **npm** always comes with a Node.js installation, be sure to update the version to the most recent with the following command.

```
npm install npm -g
```

Creating a React App - Considerations

- Including dependencies (React, React-DOM libraries, etc.)
- Making code compatible with browsers that only support older versions of JavaScript
- Transforming JSX into JavaScript
- Modularity: implementing modules in separate files, bundling them as dependencies

Creating a React App with Node.js - Setup

- Fortunately, there exists a package (through npm) that takes all of the above into consideration when creating a React app
  - Incorporates Babel for JSX and ES6 transformation
  - Incorporates Webpack for bundling

```
npm install -g create-react-app
```

Creating a React App with Node.js - Setup

- Fortunately, there exists a package (through npm) that takes all of the above into consideration when creating a React app
  - Incorporates Babel for JSX and ES6 transformation
  - Incorporates Webpack for bundling

```
npm install -g create-react-app
```

- To create new React app, run the following command in the desired parent directory of new application

```
create-react-app my-app
```

Anatomy of a React App

- **`package.json`:** information about app, lists of dependencies, shortcuts for scripts
- **`public`:** directory containing HTML files, images, other static web content
- **`src`:** directory containing JavaScript and CSS files

Starting React App with Node.js

- You can start the default app as follows:

```
cd my-app/
npm start
```

Starting React App with Node.js

- You can start the default app as follows:

```
cd my-app/
npm start
```

- This will start a web server that listens for incoming HTTP requests on port 3000 on your computer

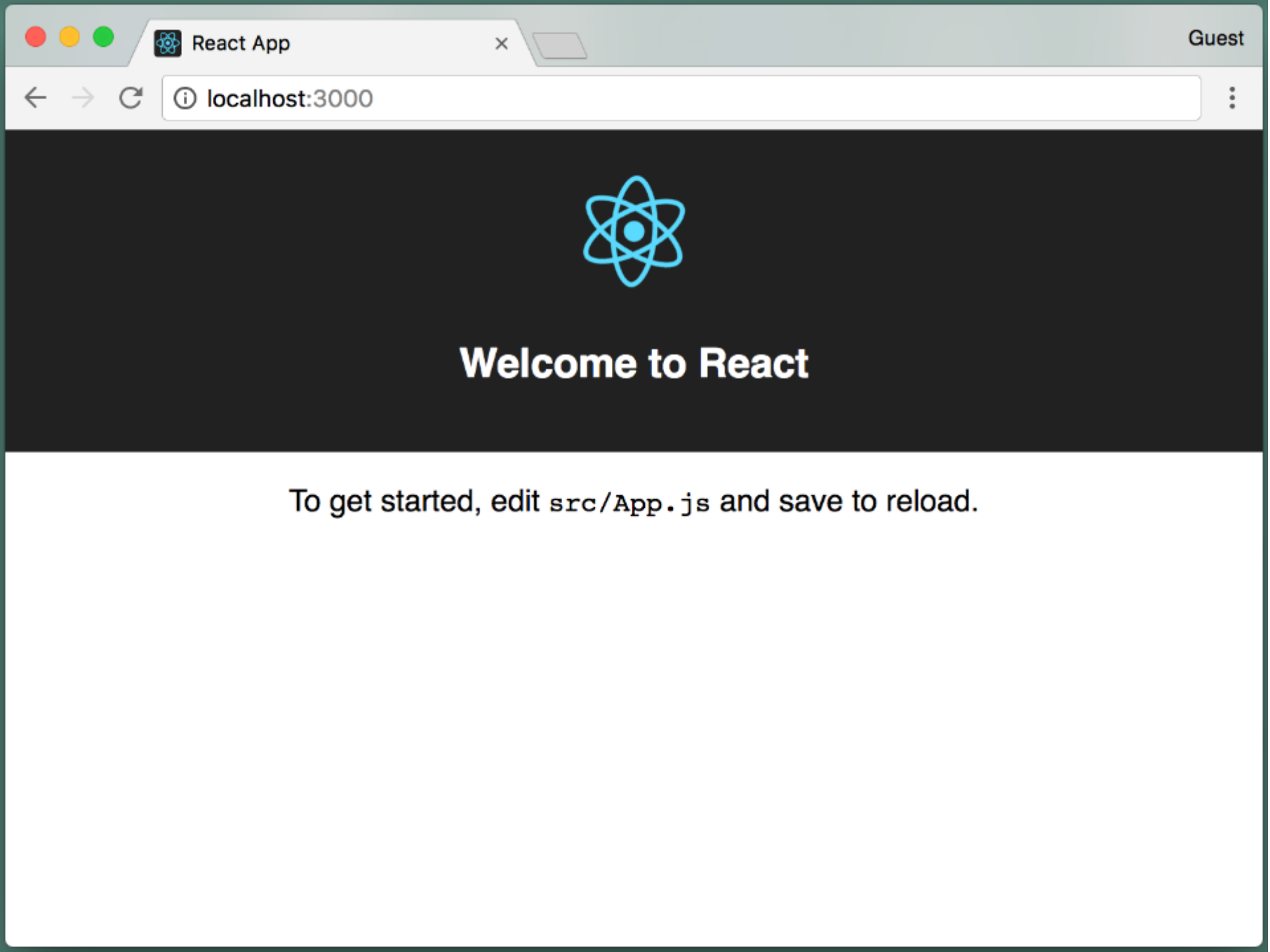
Starting React App with Node.js

- You can start the default app as follows:

```
cd my-app/
npm start
```

- This will start a web server that listens for incoming HTTP requests on port 3000 on your computer

- You can access the web server by accessing http://localhost:3000/ from your computer


React App
Guest
localhost:3000
Welcome to React
To get started, edit src/App.js and save to reload.

Incorporating Components

- We can now create separate JavaScript files for each component.
- **`src/Counter.js`** would look like this:

```
var React = require('react');

class Counter extends React.Component {
  constructor(props) { . . . }

  incrementCount() { . . . }

  render() { . . . }
};

export default Counter;
```

Incorporating Components

- We can now create separate JavaScript files for each component.
- **`src/Counter.js`** would look like this:

```
var React = require('react');

class Counter extends React.Component {
  constructor(props) { . . . }

  incrementCount() { . . . }

  render() { . . . }
};

export default Counter;
```

Incorporating Components

- We can now create separate JavaScript files for each component.
- **`src/Counter.js`** would look like this:

```
var React = require('react');

class Counter extends React.Component {
  constructor(props) { . . . }

  incrementCount() { . . . }

  render() { . . . }
};

export default Counter;
```

Incorporating Components

- We can now create separate JavaScript files for each component.
- **`src/Counter.js`** would look like this:

```
var React = require('react');

class Counter extends React.Component {
  constructor(props) { . . . }

  incrementCount() { . . . }

  render() { . . . }
};

export default Counter;
```

Incorporating Components into the App

- Edit **`src/App.js`** as follows:

```
import Counter from './Counter.js';
```

```
class App extends Component {
  render() {
   return (
      <div className="App">
         <Counter />
       </div>
   );
  }
}
```

Incorporating Components into the App

- Edit **`src/App.js`** as follows:

```
import Counter from './Counter.js';
```

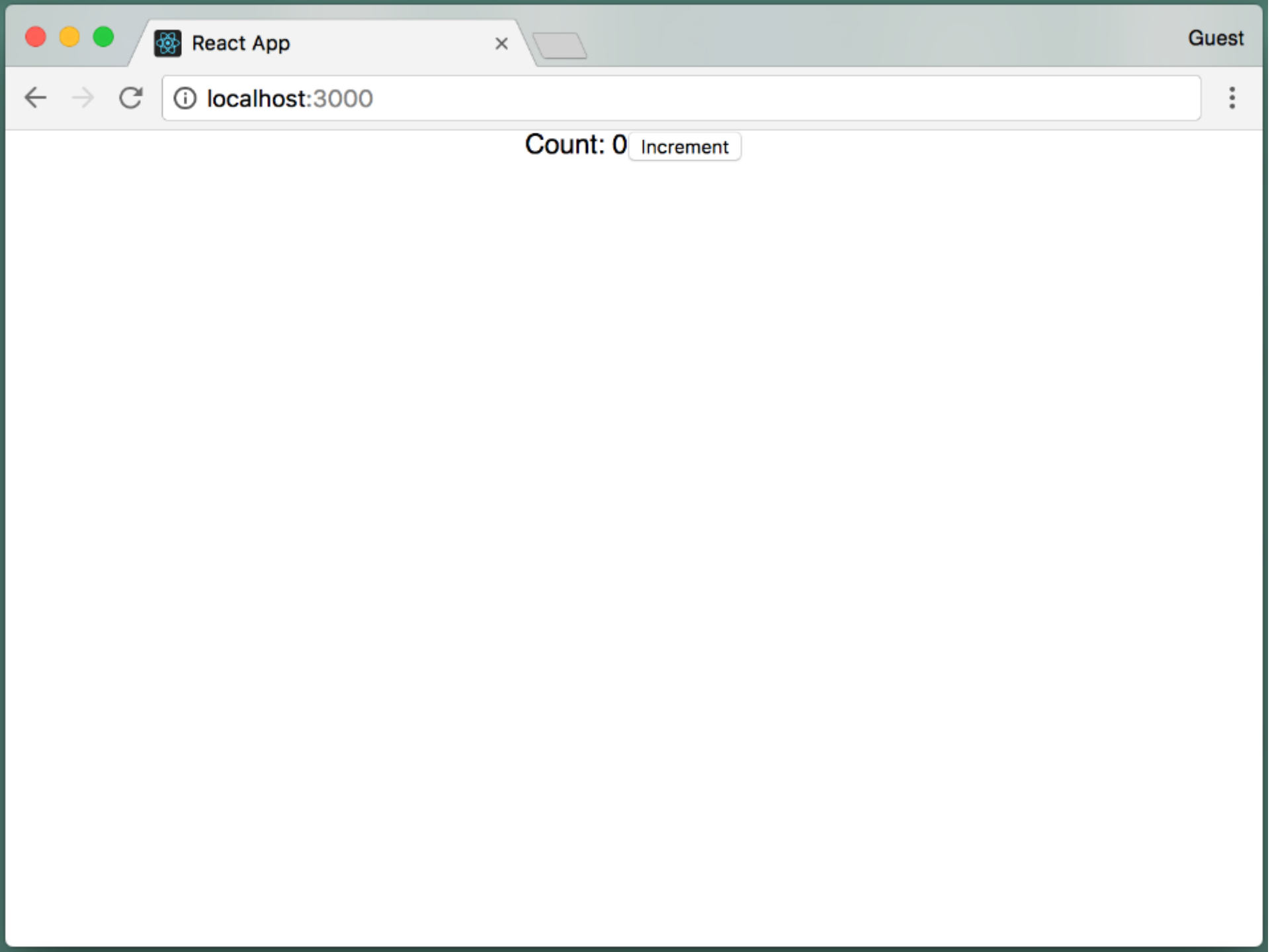
```
class App extends Component {
  render() {
   return (
      <div className="App">
         <Counter />
       </div>
   );
  }
}
```

Incorporating Components into the App

- Edit **`src/App.js`** as follows:

```
import Counter from './Counter.js';
```

```
class App extends Component {
  render() {
   return (
      <div className="App">
         <Counter />
       </div>
   );
  }
}
```


React App
Guest
localhost:3000
Count: 0
Increment

React Dogs

Guest

localhost:3000

Good Dogs

Princess: Corgi X

Riley: Husky X

Add Dog

Name

Image

Breed

Submit

React Dogs

Guest

localhost:3000

Good Dogs

Princess: Corgi X

Riley: Husky X

Add Dog

Name

Cooper

Image

https://upload.wikimedia

Breed

Catahoula Leopard

Submit

React Dogs

Guest

localhost:3000

Good Dogs

Princess: Corgi X

Riley: Husky X

Add Dog

Name

Cooper

Image

https://upload.wikimedia

Breed

Catahoula Leopard

Submit

React Dogs

Guest

localhost:3000

Good Dogs

Princess: Corgi X

Riley: Husky X

Cooper: Catahoula Leopard X

Add Dog

Name

Cooper

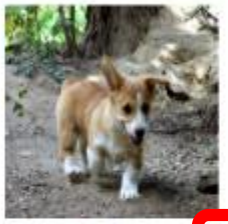
Image

React Dogs

Guest

localhost:3000

Good Dogs



Princess: Corg X

Riley: Husky X

Cooper: Catahoula Leopard X

Add Dog

Name

Cooper

Image

Good Dogs

Riley: Husky X

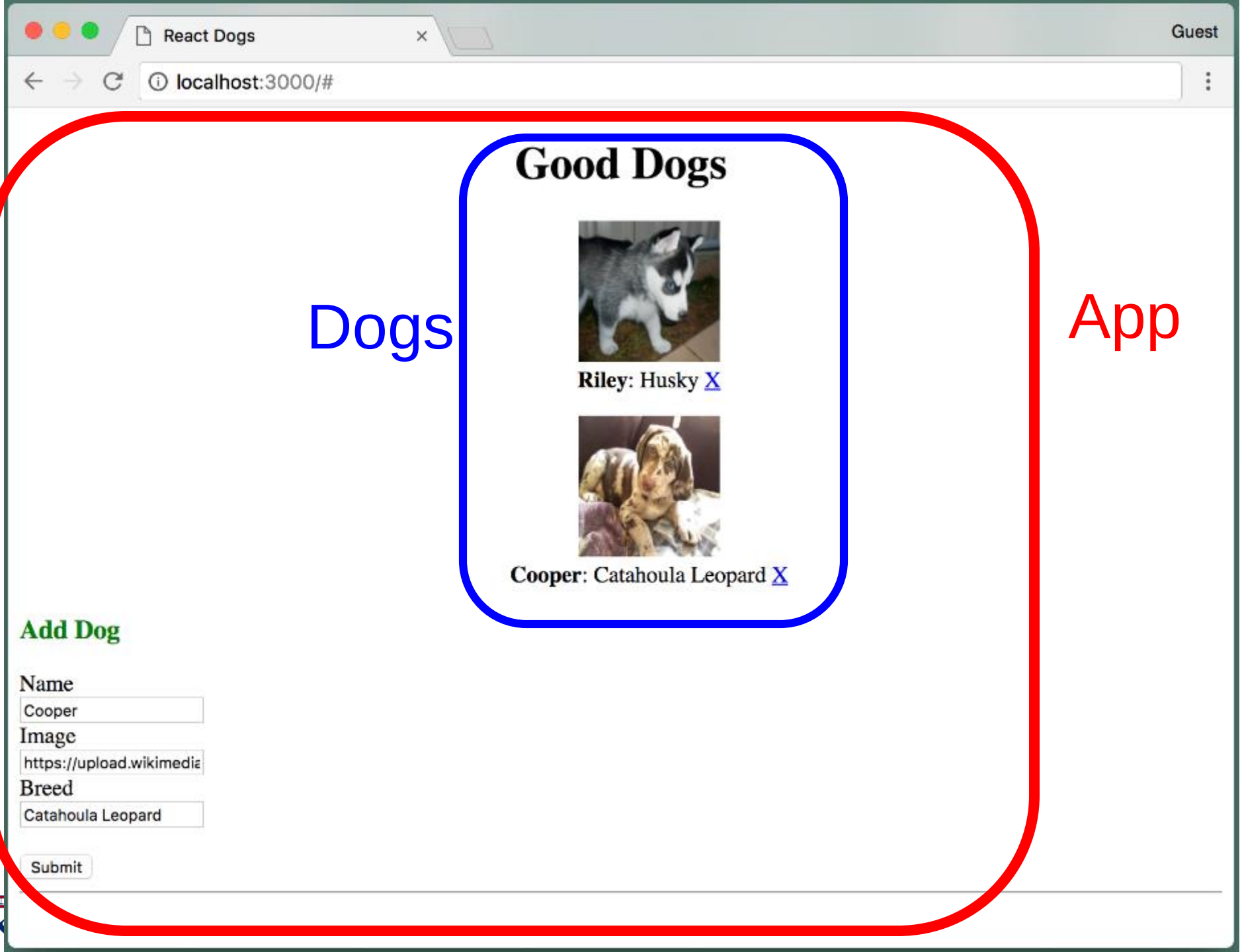
Cooper: Catahoula Leopard X

Add Dog

Name
Cooper
Image
https://upload.wikimedia
Breed
Catahoula Leopard

Submit

React Dogs
Guest
localhost:3000/#
Good Dogs
Riley: Husky X
Cooper: Catahoula Leopard X
Add Dog
Name
Cooper
Image
https://upload.wikimedia
Breed
Catahoula Leopard
Submit
App


React Dogs
Guest
localhost:3000/#
Good Dogs
Riley: Husky X
Cooper: Catahoula Leopard X
Dogs
App
Add Dog
Name
Cooper
Image
https://upload.wikimedia
Breed
Catahoula Leopard
Submit

React Dogs

Guest

localhost:3000/#

App

Dogs

DogItem

DogItem

Good Dogs

Riley: Husky X

Cooper: Catahoula Leopard X

Add Dog

Name

Cooper

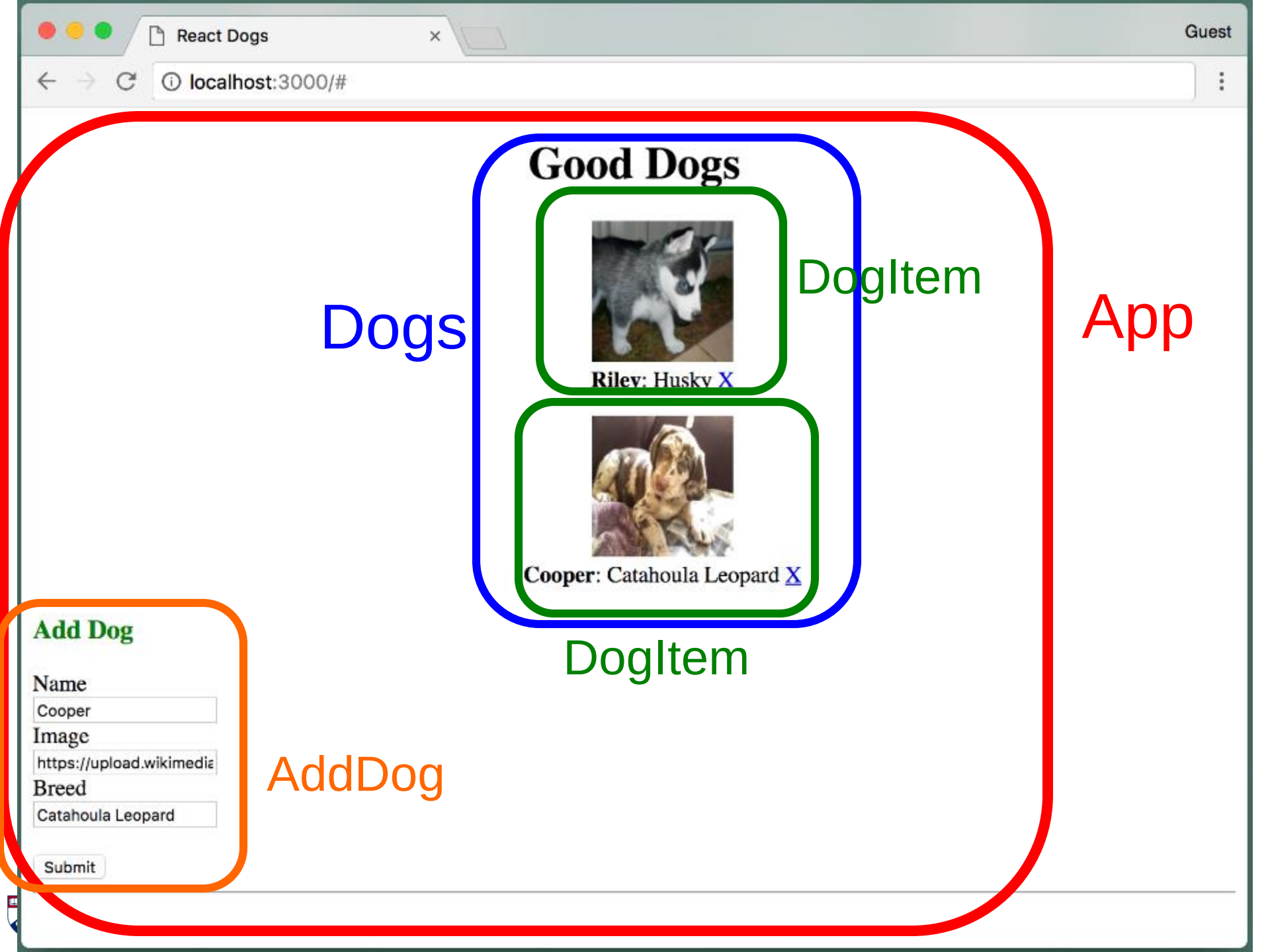
Image

https://upload.wikimedia

Breed

Catahoula Leopard

Submit


React Dogs
Guest
localhost:3000/#
Good Dogs
Riley: Husky X
Cooper: Catahoula Leopard X
Dogs
DogItem
DogItem
App
Add Dog
Name
Cooper
Image
https://upload.wikimedia
Breed
Catahoula Leopard
Submit
AddDog

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```


React Dogs
Guest
localhost:3000/#
Good Dogs
Riley: Husky X
Cooper: Catahoula Leopard X
Dogs
App
Add Dog
Name
Cooper
Image
https://upload.wikimedia
Breed
Catahoula Leopard
Submit
AddDog

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

```
import React, { Component } from 'react';
import Dogs from './components/Dogs';
import AddDog from './components/AddDog';
import './App.css';

class App extends Component {
  constructor() { . . . }

  handleAddDog(dog) { . . . }

  handleDeleteDog(name) { . . . }

  render() {
     return (
      <div className="App">
      <Dogs dogs={this.state.dogs}
            onDelete={this.handleDeleteDog.bind(this)} />
      <AddDog onAddDog={this.handleAddDog.bind(this)} />
      <hr />
      </div>
    );
   }
};
```

Testing React Apps

- **Mocha** – widely used test runner (testing framework) used to run JavaScript tests
- **Chai** – assertion library for Behavior Driven Testing
- **Enzyme** – testing utility for React for manipulating and inspecting React Component state and output

Getting Started - Installation

- To include Enzyme and Chai as dependencies, run the following command:

```
npm install --save-dev enzyme react-test-renderer chai
```

Getting Started - Installation

- To include Enzyme and Chai as dependencies, run the following command:

```
npm install --save-dev enzyme react-test-renderer chai
```

- Note that the default file structure places all JavaScript and CSS code in the '`src`' folder.
- We will create an additional folder within '`src`' named '`tests`' in which we include all testing scripts
- All test files must be in the form of `*.test.js`, e.g. **`Dogs.test.js`**

Getting Started

- Node.js should create a default `App.test.js`, or you can write your own

Getting Started

- Node.js should create a default **`App.test.js`**, or you can write your own
- Include libraries necessary for testing
  - Import React and ReactDOM for component manipulation

```
import React from 'react';

import ReactDOM from 'react-dom';
```

  - Import keywords from Enzyme

```
import { mount, shallow } from 'enzyme';
```

  - Import keywords from Chai

```
import {expect} from 'chai';
```

Getting Started

- Node.js should create a default **`App.test.js`**, or you can write your own
- Include libraries necessary for testing
  - Import React and ReactDOM for component manipulation

```
import React from 'react';

import ReactDOM from 'react-dom';
```

  - Import keywords from Enzyme

```
import { mount, shallow } from 'enzyme';
```

  - Import keywords from Chai

```
import {expect} from 'chai';
```

Getting Started

- Node.js should create a default **`App.test.js`**, or you can write your own
- Include libraries necessary for testing
  - Import React and ReactDOM for component manipulation

```
import React from 'react';

import ReactDOM from 'react-dom';
```

  - Import keywords from Enzyme

```
import { mount, shallow } from 'enzyme';
```

  - Import keywords from Chai

```
import {expect} from 'chai';
```

Getting Started

- Node.js should create a default **`App.test.js`**, or you can write your own
- Include libraries necessary for testing
  - Import React and ReactDOM for component manipulation

```
import React from 'react';

import ReactDOM from 'react-dom';
```

  - Import keywords from Enzyme

```
import { mount, shallow } from 'enzyme';
```

  - Import keywords from Chai

```
import {expect} from 'chai';
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Anatomy of a React Test

```
import React from 'react';
import { expect } from 'chai';
import { mount, shallow } from 'enzyme';
import App from '../App';

describe("Test suite for App component", function(){

  it("only one element in App class", function() {
      const wrapper = shallow(<App />);
      expect(wrapper.find(".App")).length(1);
  });

});
```

Testing React Component Relationships

```
it('Dog List contains two dogs', function() {
  const wrapper = mount(<App/>);
  expect(wrapper.find('Dogs')
                .find('DogItem')).length(2);
});
```

Testing React Component Relationships

```
it('Dog List contains two dogs', function() {
  const wrapper = mount(<App/>);
  expect(wrapper.find('Dogs')
                .find('DogItem')).length(2);
});
```

Testing React Component Relationships

```
it('Dog List contains two dogs', function() {
  const wrapper = mount(<App/>);
  expect(wrapper.find('Dogs')
                .find('DogItem')).length(2);
});
```

Testing React Component Relationships

```
it('Dog List contains two dogs', function() {
  const wrapper = mount(<App/>);
  expect(wrapper.find('Dogs')
                .find('DogItem')).length(2);
});
```

Testing React Component Relationships

```
it('Dog List contains two dogs', function() {
  const wrapper = mount(<App/>);
  expect(wrapper.find('Dogs')
                .find('DogItem')).length(2);
});
```

Testing React Component Relationships

```
it('Dog List contains two dogs', function() {
  const wrapper = mount(<App/>);
  expect(wrapper.find('Dogs')
                .find('DogItem')).length(2);
});
```

Testing React Component Relationships

```
it('Dog List contains two dogs', function() {
  const wrapper = mount(<App/>);
  expect(wrapper.find('Dogs')
                .find('DogItem')).length(2);
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
    function() {
      const wrapper = mount(<App/>);
      const adddog = wrapper.find('AddDog');

      adddog.find('#dogName').get(0).value = 'Lola';
      adddog.find('#imageURL').get(0).value = 'https://
              static.pexels.com/photos/54386/pexels-
              photo-54386.jpeg';
      adddog.find('#dogBreed').get(0).value = 'Beagle';

      const form = adddog.find('form');
      form.simulate('submit');
      expect(wrapper.find('Dogs')
                    .find('DogItem')).length(3);
      expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Data Entry and Form Submission

```
it("successfully adds dog to list when form submitted",
   function() {
     const wrapper = mount(<App/>);
     const adddog = wrapper.find('AddDog');

     adddog.find('#dogName').get(0).value = 'Lola';
     adddog.find('#imageURL').get(0).value = 'https://
             static.pexels.com/photos/54386/pexels-
             photo-54386.jpeg';
     adddog.find('#dogBreed').get(0).value = 'Beagle';

     const form = adddog.find('form');
     form.simulate('submit');
     expect(wrapper.find('Dogs')
                   .find('DogItem')).length(3);
     expect(wrapper.state().dogs[2].name == 'Lola');
});
```

Testing Links

```
it('removes dog from list when deleted', function() {
    const wrapper = mount(<App/>);
    const deleteLink = wrapper.find('a').first();

    deleteLink.simulate('click');

    expect(wrapper.find('Dogs')
                  .find('DogItem')).length(1);
});
```

Testing Links

```
it('removes dog from list when deleted', function() {
    const wrapper = mount(<App/>);
    const deleteLink = wrapper.find('a').first();

    deleteLink.simulate('click');

    expect(wrapper.find('Dogs')
                  .find('DogItem')).length(1);
});
```

Testing Links

```
it('removes dog from list when deleted', function() {
    const wrapper = mount(<App/>);
    const deleteLink = wrapper.find('a').first();

    deleteLink.simulate('click');

    expect(wrapper.find('Dogs')
                  .find('DogItem')).length(1);
});
```

Testing Links

```
it('removes dog from list when deleted', function() {
    const wrapper = mount(<App/>);
    const deleteLink = wrapper.find('a').first();

    deleteLink.simulate('click');

    expect(wrapper.find('Dogs')
                  .find('DogItem')).length(1);
});
```

Testing Links

```
it('removes dog from list when deleted', function() {
    const wrapper = mount(<App/>);
    const deleteLink = wrapper.find('a').first();

    deleteLink.simulate('click');

    expect(wrapper.find('Dogs')
                  .find('DogItem')).length(1);
});
```

Testing Links

```
it('removes dog from list when deleted', function() {
    const wrapper = mount(<App/>);
    const deleteLink = wrapper.find('a').first();

    deleteLink.simulate('click');

    expect(wrapper.find('Dogs')
                  .find('DogItem')).length(1);
});
```

Running Tests

- To run tests, navigate to the project within the terminal and run the following command:

```
npm run test
```

```
FAIL src/test/AddDog.test.js
  ● <AddDog/> › successfully adds dog to list when form submitted

    AssertionError: expected { Object (component, root, ...) } to have a length of 4 but got 3

      at Object.<anonymous> (src/test/AddDog.test.js:36:85)

FAIL src/test/App.test.js
  ● <App/> - loads › Dog List contains 3 dogs

    AssertionError: expected { Object (component, root, ...) } to have a length of 3 but got 2

      at Object.<anonymous> (src/test/App.test.js:36:85)

FAIL src/test/Dogs.test.js
  ● <Dogs/> › removes dog from list when deleted

    AssertionError: expected { Object (component, root, ...) } to have a length of 2 but got 1

      at Object.<anonymous> (src/test/Dogs.test.js:25:72)

Test Suites: 3 failed, 3 total
Tests:       3 failed, 5 passed, 8 total
Snapshots:   0 total
Time:        2.3s, estimated 3s
Ran all test suites.

Watch Usage
 › Press p to filter by a filename regex pattern.
 › Press q to quit watch mode.
 › Press Enter to trigger a test run.
```

Running Tests: Success!

```
 PASS  src/test/App.test.js
 PASS  src/test/Dogs.test.js
 PASS  src/test/AddDog.test.js

Test Suites: 3 passed, 3 total
Tests:       8 passed, 8 total
Snapshots:   0 total
Time:        2.398s
Ran all test suites.

Watch Usage
 › Press o to only run tests related to changed files.
 › Press p to filter by a filename regex pattern.
 › Press q to quit watch mode.
 › Press Enter to trigger a test run.
```

Summary

- We can use **Node.js** to create React applications
- This allows us to put component code into separate .js files and then include them into our App as necessary
- **Mocha**, **Chai**, and **Enzyme** can be used for testing our React apps