



Advertisement

CODE > REACT

Understanding Nested Routing in React

by [Roy Agasthyan](#) 31 Jan 2017Difficulty: Intermediate Length: Short Languages: English ▼

React

JavaScript

HTML

Web Development



React is a JavaScript library built by Facebook for building user interfaces. If you are new to React, I would recommend reading one of our earlier tutorials on [getting started with React](#). In this tutorial, we'll discuss how to handle nested routing in a React web application.

Setting Up the App

We'll start by setting up our React web application. Let's create a project folder called `ReactRouting`. Inside `ReactRouting`, create a file called `app.html`. Below is the `app.html` file:

```

01 | <html>
02 |
03 | <head>
04 |   <title>TutsPlus - React Nested Routing</title>
05 | </head>
06 |
07 | <body>
08 |   <div id="app"></div>
09 |   <script src="bundle.js"></script>
10 | </body>
11 |
12 | </html>

```

Inside the `ReactRouting` folder, initialize the project using Node Package Manager (npm).

```
1 | npm init
```

Enter the project-specific details or press **Enter** for defaults, and you should have the `package.json` file inside the `ReactRouting` folder.

```

01 | {
02 |   "name": "reactroutingapp",
03 |   "version": "1.0.0",
04 |   "description": "",
05 |   "main": "app.js",
06 |   "scripts": {
07 |     "test": "echo \"Error: no test specified\" && exit 1"
08 |   },
09 |   "author": "",
10 |   "license": "ISC"
11 | }

```

Install the `react` and `react-dom` dependencies using npm.

```
1 | npm install react react-dom --save
```

Install the `babel` package using `npm` and save it to the `package.json` file.

```
1 | npm install --save-dev webpack-dev-server webpack babel-core babel-loader babel-prese
```

`babel` packages are required to transform the JSX code into JavaScript. Create a webpack config file to handle the webpack configurations. Here is how the `webpack.config.js` file would look:

```

01 module.exports = {
02   entry: './app.js',
03
04   output: {
05     filename: 'bundle.js',
06   },
07
08   module: {
09     loaders: [
10       { exclude: /node_modules/, loader: 'babel-loader?presets[]=es2015&presets[]=re
11     ]
12   }
13 }

```

`app.js` is the file where our React code would reside. Create a file called `app.js` inside the `ReactRouting` folder. Import the required react libraries in `app.js`. Create a component called `App` having a div with some text. Render the component using the render method. Here is how the basic `app.js` file would look:

```

01 import React from 'react'
02 import { render } from 'react-dom'
03
04 const App = () => {
05   return (
06     <h2>{'TutsPlus - Welcome to React Nested Routing App'}</h2>
07   );
08 };
09
10 render((
11   <App />
12 ), document.getElementById('app'))

```

Restart the `webpack` development server.

```
1 | webpack-dev-server
```

Once the server has been restarted, you should be able to view the React app running on <http://localhost:8080/>.

Handling Routing in React

We'll be making use of `react-router` to implement routing and nested routing. First, install `react-router` in our project.

```
1 | npm install react-router --save
```

Now we have `react-router` installed in our project. Let's create a couple of views to implement routing. Inside the `app.js` file, create two components called `view1` and `view2`. Here is how the code would look:

```

01  const View1 = () => {
02      return(
03          <h3>{'Welcome to View1'}</h3>
04      );
05  };
06
07  const View2 = () => {
08      return(
09          <h3>{'Welcome to View2'}</h3>
10      );
11  };

```

Create a `Menu` component to display `View1` and `View2` on clicking. Here is how it looks:

```

01  const Menu = () => {
02      return(
03          <section>
04
05              <ul>
06                  <li>{'Home'}</li>
07                  <li>
08                      <Link to="/view1">{'View1'}</Link>
09                  </li>
10                  <li>
11                      <Link to="/view2">{'View2'}</Link>
12                  </li>
13              </ul>
14          </section>
15      )
16  };

```

Let's use the react router to display the default route for our application. Make the `Menu` component the default route. Also define the route for the `View1` and `View2` components. Here is how it looks:

```

1  render(
2      <Router history={hashHistory}>
3          <Route component={Menu} path="/"></Route>
4          <Route component={View1} path="/view1"></Route>
5          <Route component={View2} path="/view2"></Route>
6      </Router>,
7      document.getElementById('app'));

```

Save the above changes and restart the `webpack` server. You should have the vertical navigation menu displayed at <http://localhost:8080/webpack-dev-server/>.

App ready.

- Home
- [View1](#)
- [View2](#)

Try clicking on the `view1` and `view2` links, and it should navigate to the appropriate components.

Advertisement

Nested Routes in React

We already have an `App` component which displays the header text. Suppose we want the menu click to display the appropriate view, but we want it to be rendered inside the `App` component.

We want the navigation menu to be available across all pages. In order to achieve this, we'll try to nest our react components `View1` and `View2` inside the `App` component. So whenever the `View1` is rendered, it's displayed inside the `App` component.

Now modify the router config and place the `View1` and `View2` route config inside the `App` component route.

```
1 render(  
2   <Router history={hashHistory}>  
3     <Route component={App} path="/">  
4       <Route component={View1} path="/view1"></Route>  
5       <Route component={View2} path="/view2"></Route>  
6     </Route>  
7   </Router>,  
8   document.getElementById('app'));
```

Modify the `App` component to nest the `Menu` component inside it.

```
1 const App = () => {  
2   return (  
3     <div>  
4       <h2>{'TutsPlus - Welcome to React Nested Routing App'}</h2>  
5       <Menu />  
6     </div>  
7   );  
8 };
```

To render the nested routes inside the `App` component, we'll add `props.children` to the `App` component. Pass in `props` as a function argument to the `App` component.

```
1 const App = (props) => {  
2   return (  
3     <div>  
4       <h2>{'TutsPlus - Welcome to React Nested Routing App'}</h2>  
5       <Menu />  
6       {props.children}  
7     </div>  
8   );  
9 };
```

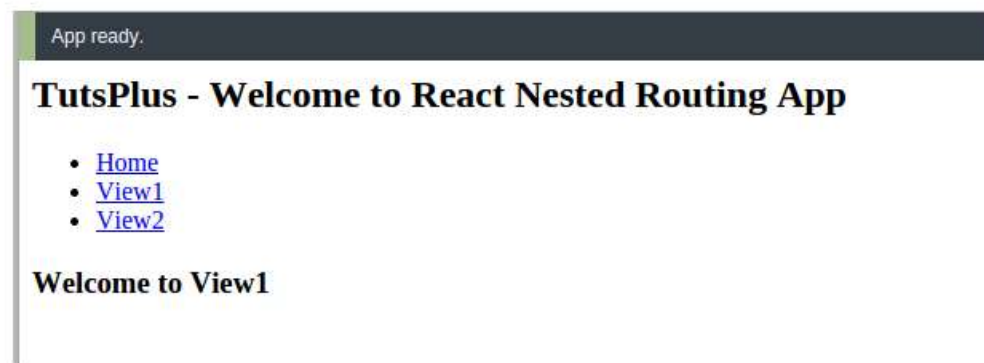
Let's also add one `Home` component to our application.

```
1 const Home = () => {  
2   return(  
3     <div>  
4       <h3>{'Welcome Home !!'}</h3>  
5     </div>  
6   );  
7 };
```

Include the `Home` component in the nested route list.

```
1 render(  
2   <Router history={hashHistory}>  
3     <Route component={App} path="/">  
4       <Route component={Home} path="/home"></Route>  
5       <Route component={View1} path="/view1"></Route>  
6       <Route component={View2} path="/view2"></Route>  
7     </Route>  
8   </Router>,  
9   document.getElementById('app'));
```

Save the above changes and restart the server, and you should be able to view the application. Click on the menu items listed, and each when clicked should render nested inside the `App` component.



Wrapping It Up

In this tutorial, you saw how to implement nested routing in a React application using `react-router`. You can develop small independent components and then use the components to create larger applications. Making it possible to stitch up small components into large applications is what makes React really powerful.

React is becoming one of the de facto platforms for working on the web. It's not without its learning curves, and there are plenty of resources to keep you busy, as well. If you're looking for additional resources to study or to use in your work, check out what we have available in the [Envato marketplace](https://code.tutsplus.com/tutorials/understanding-nested-routing-in-react--cms-27955).

Source code from this tutorial is available on [GitHub](#). Let us know your suggestions and thoughts in the comments below.

Advertisement



Roy Agasthyan

I'm a software engineer by profession and love to write tutorials and educational stuff.

FEED LIKE FOLLOW FOLLOW

Weekly email summary

Subscribe below and we'll send you a weekly email summary of all new Code tutorials. Never miss out on learning about the next big thing.

Update me weekly

Advertisement

Translations

Envato Tuts+ tutorials are translated into other languages by our community members—you can be involved too!

Translate this post

Powered by  native

1 Comment Tuts+ Hub

1 Login ▾

♥ Recommend 50

🐦 Tweet

f Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

**Treefish Zhang** • 2 years ago

The syntax of nested routes seems to have changed since then. I got error 'You should not use `<route component="">` and `<route children="">` in the same route; `<route children="">` will be ignored', and the fix to that is in (<https://stackoverflow.com/q...>

In addition, I needed to change `app.html` to `index.html`; during `npm init` needed to name 'app.js' for 'entry' in lieu of default 'index.js'

^ | ▾ • Reply • Share ›



QUICK LINKS - Explore popular categories

ENVATO TUTS+	+
JOIN OUR COMMUNITY	+
HELP	+



tuts+

27,321	1,220	38,309
Tutorials	Courses	Translations

Envato.com Our products Careers Sitemap

© 2019 Envato Pty Ltd. Trademarks and brands are the property of their respective owners.

Follow Envato Tuts+