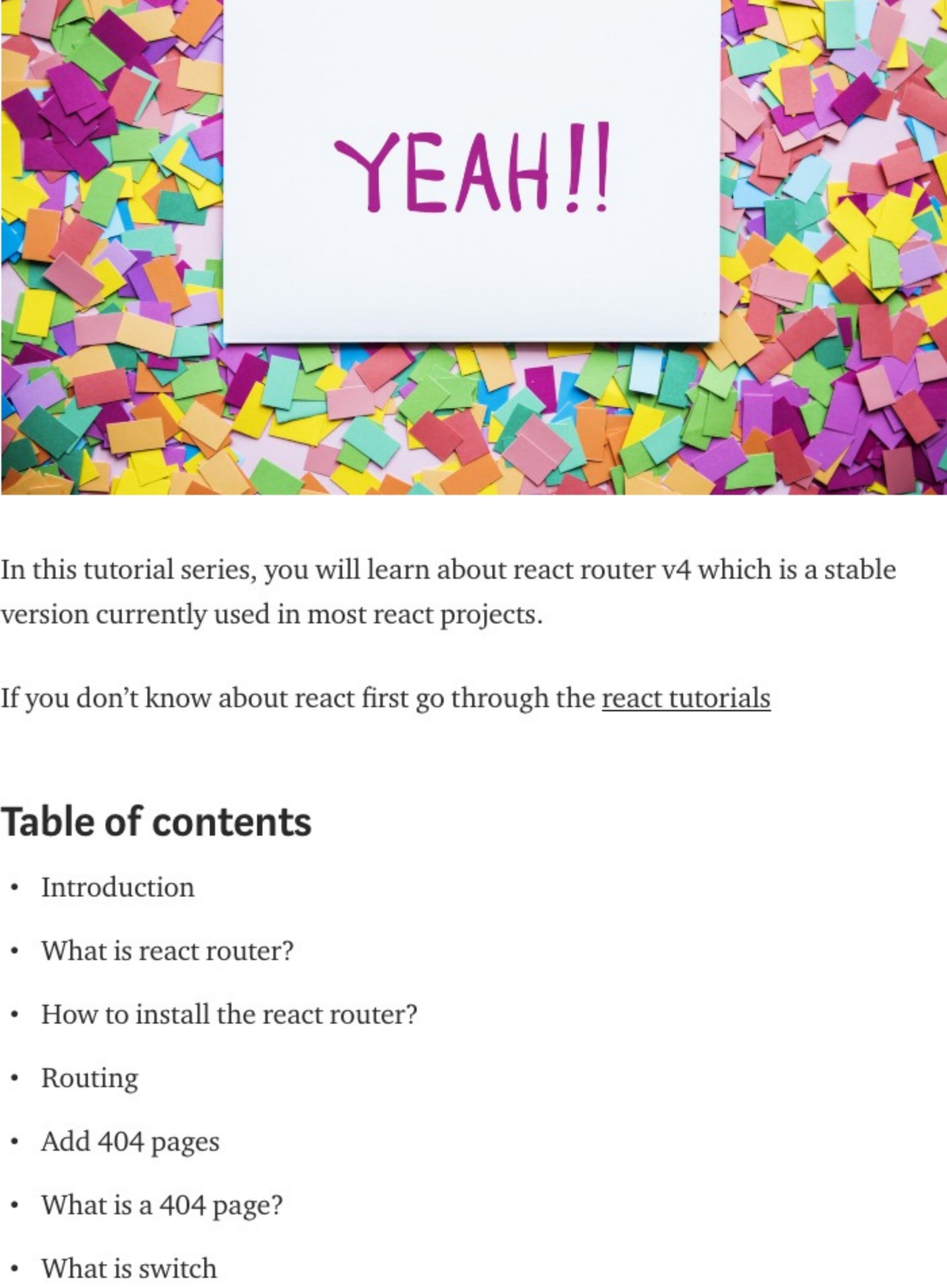


Getting started with React Router



Sai gowtham [Follow](#)
Sep 17, 2018 · 9 min read



In this tutorial series, you will learn about react router v4 which is a stable version currently used in most react projects.

If you don't know about react first go through the [react tutorials](#)

Table of contents

- Introduction
- What is react router?
- How to install the react router?
- Routing
- Add 404 pages
- What is a 404 page?
- What is switch
- Url parameters
- Nested Routes
- Programmatically navigate

Introduction

What is react router?

React router is a routing library built on top of the react which is used to create the routing in react apps.

Is react router is static or dynamic?

- Before the react router v4 it is static after v4 it is Dynamic.

In single page apps, there is only single html page. we are reusing the same html page to render the different components based on the navigation.

But in multipage apps, you will get an entirely new page from the server when you navigate.

“If you stuck anywhere in the middle please refer to [code repository](#)”

Getting started

How to install the react router?

we are using the create-react-app to create the app.

```
npx create-react-app routing
cd routing
```

To install the react router you need to download the react-router-dom package by running the following commands.

```
npm install react-router-dom

npm start //to run dev server
```

Now open the routing folder in your favorite code editor.

If you navigate to the public/index.html you will see a single html file which looks like this.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="theme-color" content="#000000">
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
    <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
    <title>React App</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

Currently, in our app, there is only a single App component.

Let's create two more components.

users.js

```
import React from 'react'

class Users extends React.Component {
  render() {
    return <h1>Users</h1>
  }
}

export default Users
```

contact.js

```
import React from 'react'

class Contact extends React.Component {
  render() {
    return <h1>Contact</h1>
  }
}

export default Contact
```

app.js

```
import React from 'react'
class App extends React.Component {
  render() {
    return (
      <div>
        <h1>Home</h1>
      </div>
    )
  }
}

export default App
```

Now our app has three components one is App and the other two are Users and Contact.

Routing

open the index.js file and import the three components (App,Users,Contact)

index.js

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'
import App from './App'
import Users from './users'
import Contact from './contact'

ReactDOM.render(<App />, document.getElementById('root'))
```

React router gives us three components [Route,Link,BrowserRouter] which help us to implement the routing.

index.js

```
import React from 'react'
import ReactDOM from 'react-dom'
import { Route, Link, BrowserRouter as Router } from 'react-router-dom'
import './index.css'
import App from './App'
import Users from './users'
import Contact from './contact'

const routing = (
  <Router>
    <div>
      <Route path="/" component={App} />
      <Route path="/users" component={Users} />
      <Route path="/contact" component={Contact} />
    </div>
  </Router>
)

ReactDOM.render(routing, document.getElementById('root'))
```

Let's implement the routing.

In the Route component, we need to pass the two props

- path: it means we need to specify the path.
- component: which component user needs to see when they will navigate to that path.

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'
import { Route, Link, BrowserRouter as Router } from 'react-router-dom'
import App from './App'
import Users from './users'
import Contact from './contact'

const routing = (
  <Router>
    <div>
      <Route exact path="/" component={App} />
      <Route path="/users" component={Users} />
      <Route path="/contact" component={Contact} />
    </div>
  </Router>
)

ReactDOM.render(routing, document.getElementById('root'))
```

Now if you enter manually localhost:3000/users you will see Users component is rendered.



Adding 404 Pages

What is a 404 page?

A 404 page is also called not found page it means when a user navigates to the wrong path that doesn't present in the website we need to show the not found page.

How to add a 404 page in react?

we need to render another component called Switch which is provided by the react router.

What is Switch?

Switch component helps us to render the components only when path matches otherwise it fallbacks to the not found component.

let's create a Not found component.

notfound.js

```
import React from 'react'

const Notfound = () => <h1>Not found</h1>

export default Notfound
```

index.js

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'
import { Route, Link, BrowserRouter as Router, Switch } from 'react-router-dom'
import App from './App'
import Users from './users'
import Contact from './contact'
import Notfound from './notfound'

const routing = (
  <Router>
    <div>
      <ul>
        <li>
          <Link to="/">Home</Link>
        </li>
        <li>
          <Link to="/users">Users</Link>
        </li>
        <li>
          <Link to="/contact">Contact</Link>
        </li>
      </ul>
      <Switch>
        <Route exact path="/" component={App} />
        <Route path="/users" component={Users} />
        <Route path="/contact" component={Contact} />
        <Route component={Notfound} />
      </Switch>
    </div>
  </Router>
)

ReactDOM.render(routing, document.getElementById('root'))
```

Let's check it now by manually entering wrong path *localhost:3000/posts*.

Not found