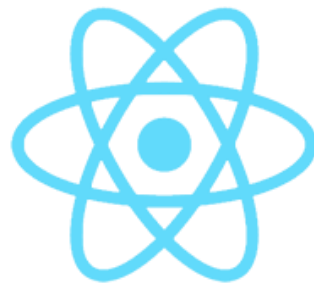




# React-Router



# React

- Setup of React-Router
- Router Configuration
- Adding Navigational Links
- Create Active Link
- Nested Routing

# Setup of React-Router

3

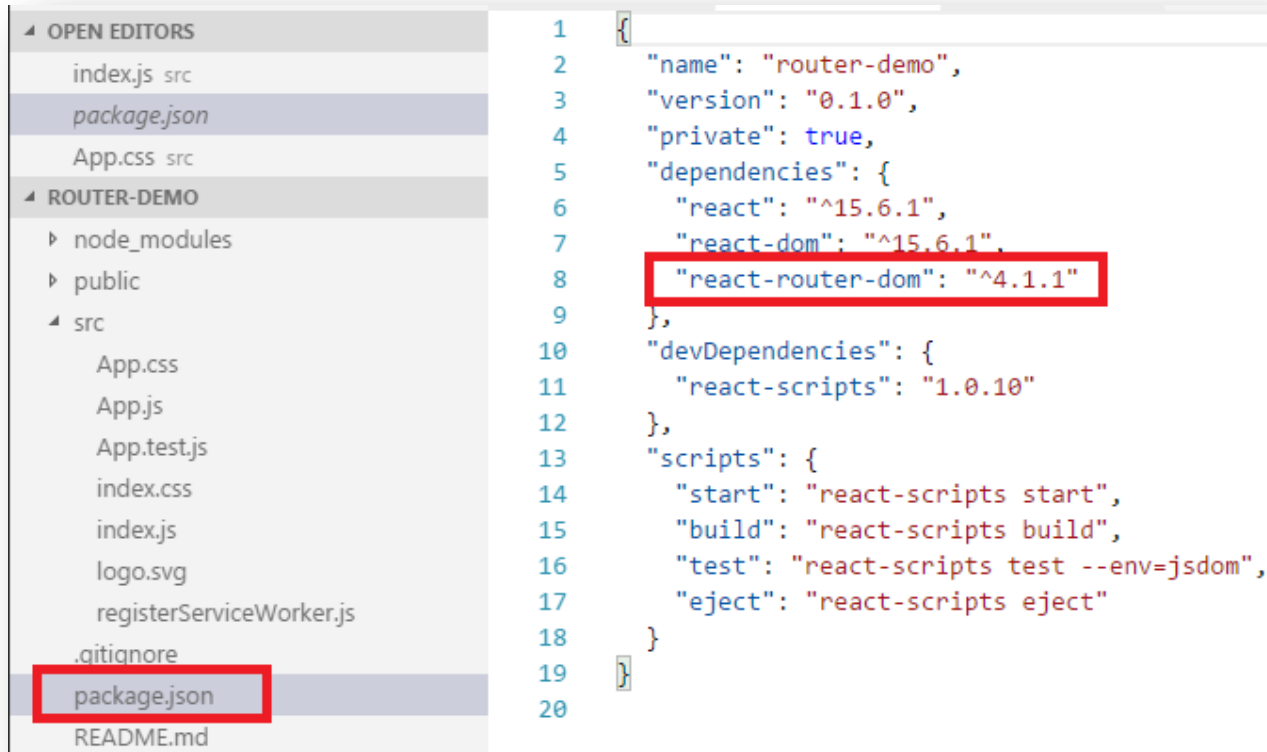
Create a new react application and name it as react-router-demo as below.

***create-react-app react-router-demo***

Now run below command to install ***react-router-dom*** in your application

***npm install react-router-dom -S***

To verify this we can open the package.json and check whether it is installed and added in package.json



```
1  {
2    "name": "router-demo",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "react": "^15.6.1",
7      "react-dom": "^15.6.1",
8      "react-router-dom": "^4.1.1"
9    },
10   "devDependencies": {
11     "react-scripts": "1.0.10"
12   },
13   "scripts": {
14     "start": "react-scripts start",
15     "build": "react-scripts build",
16     "test": "react-scripts test --env=jsdom",
17     "eject": "react-scripts eject"
18   }
19 }
```

Create a stateless component and name it as ***RouterComponent***. Import {BrowserRouter as Router, Route} from 'react-router-dom'

```
React-router-demo/src/index.js
-----
import React from 'react';
import ReactDOM from 'react-dom';
import {
  BrowserRouter as Router,
  Route,
} from 'react-router-dom';

const RouterComponent = () =>(
  {/* will be adding the code for Router configuration*/}
)

ReactDOM.render(<RouterComponent/>,document.getElementById("root"));
```

Add Router using `<Router></Router>` in RouterComponent

```
React-router-demo/src/index.js
-----
import React from 'react';
import ReactDOM from 'react-dom'; import {
  BrowserRouter as Router,
  Route,
} from 'react-router-dom';

const RouterComponent = () =>(
  <Router>
  <Route exact path="/" component={HomeComponent}/>
</Router>
)

ReactDOM.render(<RouterComponent/>,document.getElementById("root"));
```

Path : url path

Component : based on path which Component will render

Create HomeComponent and return the ***HOME*** String to represent home component rendering.

```
React-router-demo/src/index.js
-----
import React from 'react';
import ReactDOM from 'react-dom'; import {
  BrowserRouter as Router,
  Route,
} from 'react-router-dom';
const HomeComponent = () =>(
  <h1>Home Page</h1>
)
const RouterComponent = () =>(
  <Router>
  <Route exact path="/" component={HomeComponent}/>
</Router>
)

ReactDOM.render(<RouterComponent/>,document.getElementById("root"));
```

Now save the changes and check on Browser, your Home component must be rendering.

Now create another Router for about component. And create AboutComponent that should return About String.

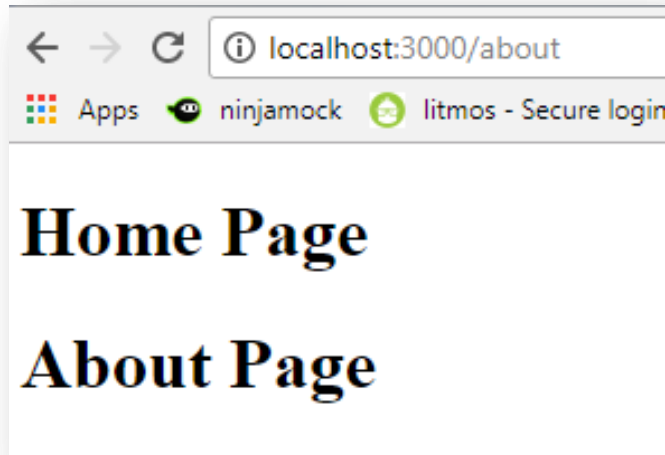
```
React-router-demo/src/index.js
-----
import React from 'react';
import ReactDOM from 'react-dom'; import {
  BrowserRouter as Router,
  Route,
} from 'react-router-dom';
const HomeComponent = () =>(
  <h1>Home Page</h1>
)
const AboutComponent = () =>(
  <h1>About Page</h1>
)
const RouterComponent = () =>(
  <Router>
    <Route exact path="/" component={HomeComponent}/>
    <Route exact path="/about" component={AboutComponent}/>
  </Router>
)
ReactDOM.render(<RouterComponent/>,document.getElementById("root"));
```

Now check your application, it will throw some error.

***It is because your <Router> returning two values, you can return only single value, so place both the Route tags in <div></div>***

Now if you check it on browser, Home Page will be displayed.

Add following in URL : *localhost:3000/about* and hit Enter. You will find below output.



It is rendering both the components, because it is following the path of Home i.e. / and also the path of about i.e. /about

To resolve this you need to tell that for Home will be rendering exactly for /  
Add below code and recheck.

```
React-router-demo/src/index.js
-----
. . . . .
<Route exact path="/" component={HomeComponent}/>
```



# Adding Navigational Links

9

Again and again hitting the browser to activate a particular view is not good practice, it is better to have Navigation Menu, where user can click and activate desired component.

Let us create Links component that will expose the Links. Add Link attribute in import

```
React-router-demo/src/index.js
-----
import React from 'react';
import ReactDOM from 'react-dom'; import {
  BrowserRouter as Router,
  Route, Link
} from 'react-router-dom';
. . HomeComponent . .
. . AboutComponent. .
const Links = () =>(
  <ul>
    <li><Link to="/">Home</Link></li>
    <li><Link to="/about">About</Link></li>
  </ul>
)
const RouterComponent = () =>(
  <Router><div>
    <Links/>
    <Route exact path="/" component={HomeComponent}/>
    <Route exact path="/about" component={AboutComponent}/>
  </Router></div>
)
ReactDOM.render(<RouterComponent/>, document.getElementById("root"));
```

Now if you check, your links should function.

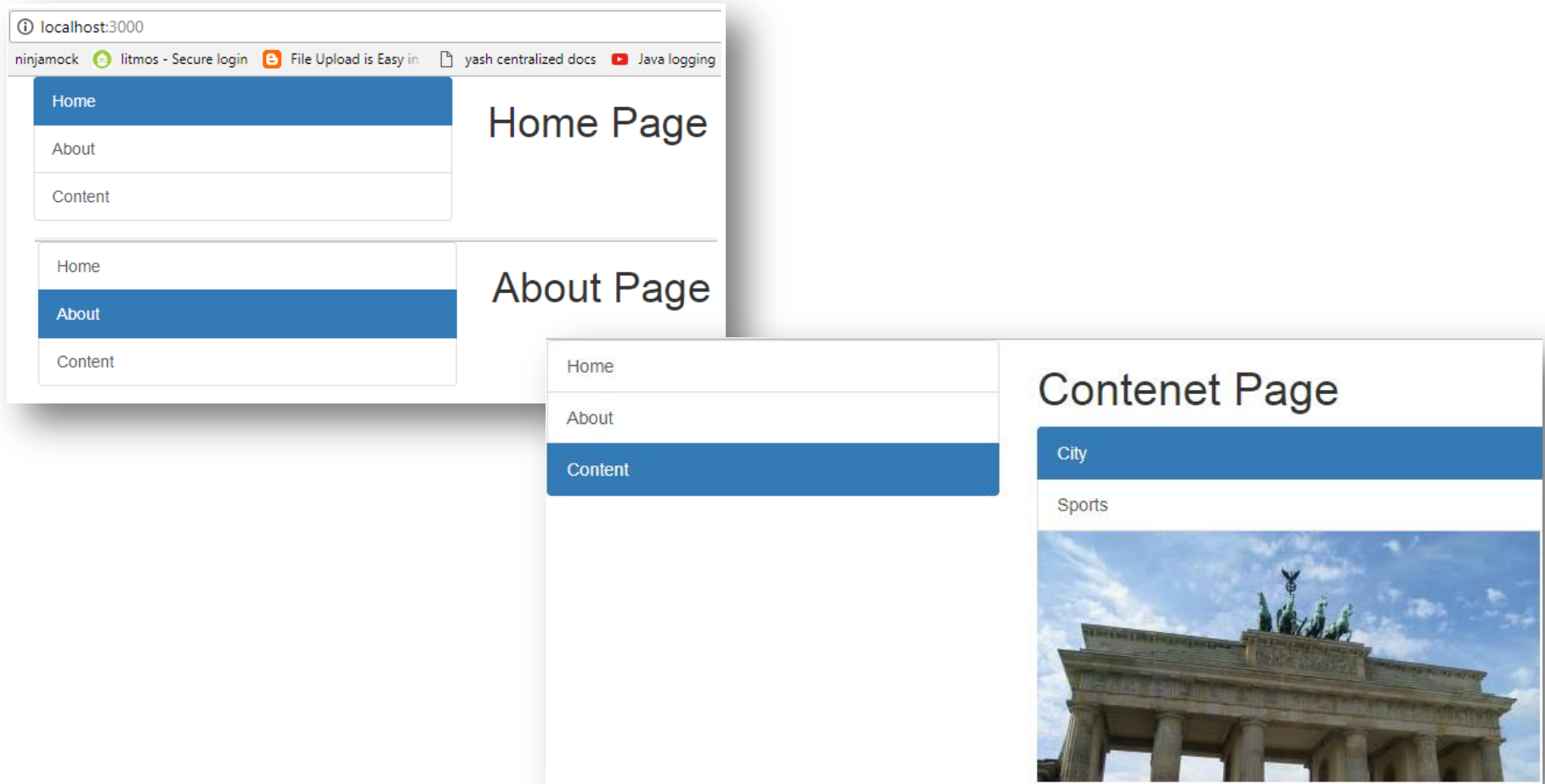
Link tag is with limited features, if you want to show the active link, you will have to use another option. i.e. NavLink. Include NavLink in import statement and make appropriate changes.

```
React-router-demo/src/index.js
-----
import React from 'react';
import ReactDOM from 'react-dom'; import {
  BrowserRouter as Router,
  Route, Link, NavLink
} from 'react-router-dom';
. . HomeComponent . .
. . AboutComponent. .
const Links = () =>(
  <ul>
  <li><NavLink activeClassName="selected" to="/">Home</NavLink></li>
  <li><NavLink activeClassName="selected" to="/about">About</NavLink></li>
  </ul>
)
. . RouterComponent . .
ReactDOM.render(<RouterComponent/>,document.getElementById("root"));
```

Your link will be activated but the problem will be that when you will click on About. It will not bring the Home link in default state. This is the same problem we faced with path in Route. We need to add exact here as well. Make below changes and check.

```
React-router-demo/src/index.js
-----
<li><NavLink exact activeClassName="selected" to="/">Home</NavLink></li>
```

Here I will show you something about Routing inside another Routing. Below screen shots will clear it.



To add some styling I have used here the bootstrap.


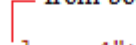
Go to below location and copy the link of bootstrap from CDN and paste that in the index.html's head section.

<http://getbootstrap.com/getting-started/#download>

```
<!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">
    <!--
      manifest.json provides metadata used when your web app is added to the
      homescreen on Android. See https://developers.google.com/web/fundamentals/engage-and-retain/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
    <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>
      You need to enable JavaScript to run this app.
    </noscript>
    <div class="container">
      <div id="root"></div>
    </div>
```

Now make appropriate changes in RouterComponent in index.js.

```
const RouterComponent = () =>(  
  <Router>  
    <div className="row">  will be implementing  
      <section className="col-sm-4">  from bootstrap  
        <Links/>  
      </section>  
      <section className="col-sm-8">  
        <Route exact path="/" component={HomeComponent}/>  
        <Route path="/about" component={AboutComponent}/>  
        <Route path="/content" component={ContentComponent}/> — one more Route configuration  
      </section>  
    </div>  
  </Router>  
)
```

Update the Links component to add bootstrap classes.

```
const Links = () =>(  
  <div className="list-group">  
    <NavLink className="list-group-item" exact activeClassName="active" to="/">Home</NavLink>  
    <NavLink className="list-group-item" activeClassName="active" to="/about">About</NavLink>  
    <NavLink className="list-group-item" activeClassName="active" to="/content">Content</NavLink>  
  </div>  
)
```

Create ContentComponent

```
const ContentComponent = () =>(
  <div>
    <h1>Content Page</h1>
  </div>
)
```

Now we want to add the functionality for the Content section.

```
const ContentComponent = () =>(
  <div>
    <h1>Content Page</h1>
    <div>
      <NavLink className="list-group-item" exact activeClassName="active" to="/content/city">City</NavLink>
      <NavLink className="list-group-item" exact activeClassName="active" to="/content/sports">Sports</NavLink>
    </div>
  </div>
)
```

Check your application

Now we want to add the Route for city and Sports, so while clicking the city or sports it should open the appropriate section.

```
const ContentComponent = () =>(  
  <div>  
    <h1>Content Page</h1>  
    <div>  
      <NavLink className="list-group-item" exact activeClassName="active" to="/content/city">City</NavLink>  
      <NavLink className="list-group-item" exact activeClassName="active" to="/content/sports">Sports</NavLink>  
      <Route path="/content/:contentName" component={ContentDetails}/>  
    </div>  
  </div>  
)
```

Create ContentDetails component.

```
const ContentDetails = (props) =>(  
  <div>{props.match.params.contentName}</div>  
)
```

Now we would like to show the some city and some sports image.

```
const ContentDetails = (props) =>(  
  <div>{props.match.params.contentName  
    ?  
    <div>  
      <img src={'http://lorempixel.com/400/200/' + props.match.params.contentName + '/1/'}/></div>  
      : null }  
    </div>  
  </div>  
)
```

Now check your application



Thank You!

Email: [info@yash.com](mailto:info@yash.com)

Web: [www.yash.com](http://www.yash.com)

© YASH Technologies, 1996-2013. All rights reserved.

The information in this document is based on certain assumptions and as such is subject to change. No part of this document may be reproduced, stored or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of YASH Technologies Inc. This document makes reference to trademarks that may be owned by others. The use of such trademarks herein is not as assertion of ownership of such trademarks by YASH and is not intended to represent or imply the existence of an association between YASH and the lawful owners of such trademarks.

Presented By : Pankaj Sharma