

Url Parameters

Url parameters helps us to render the same component based on its dynamic url like in our Users component assume that they are different users with id 1,2,3.

```
<Route path="users/:id" component={Users} />
```

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/:id" component={Users} />
        <Route path="/contact" component={Contact} />
        <Route component={NotFound} />
      </Switch>
    </div>
  </Router>
)

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

Open users.js file and add console.log(this.props).

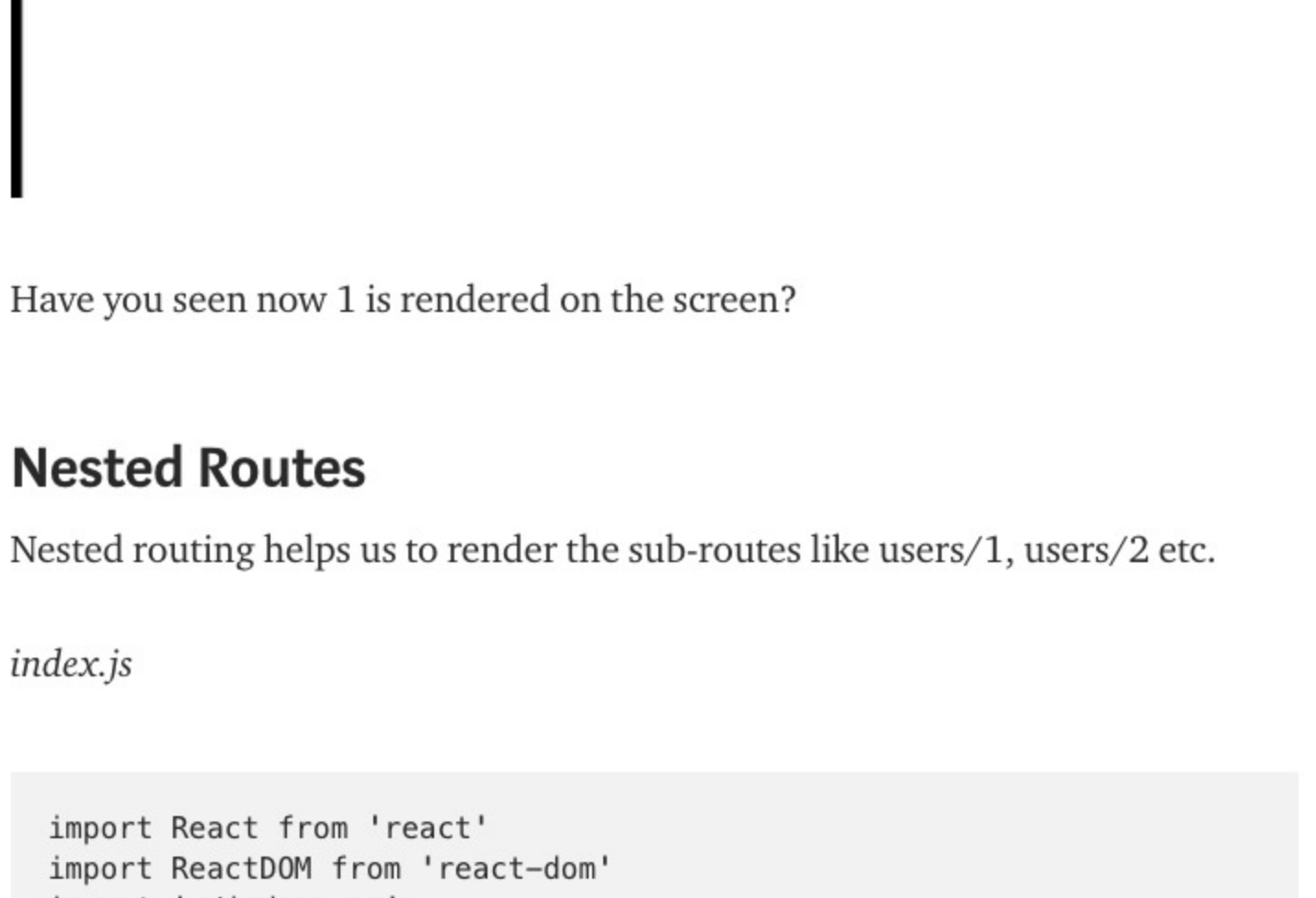
users.js

```
import React from 'react'

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

export default Users
```

Now open your browser and manually type this url localhost:3000/users/1 you will see an object in your console which is passed by the react router.



Have you seen the params object with id 1 the same thing we passed in the url localhost:3000/users/1.

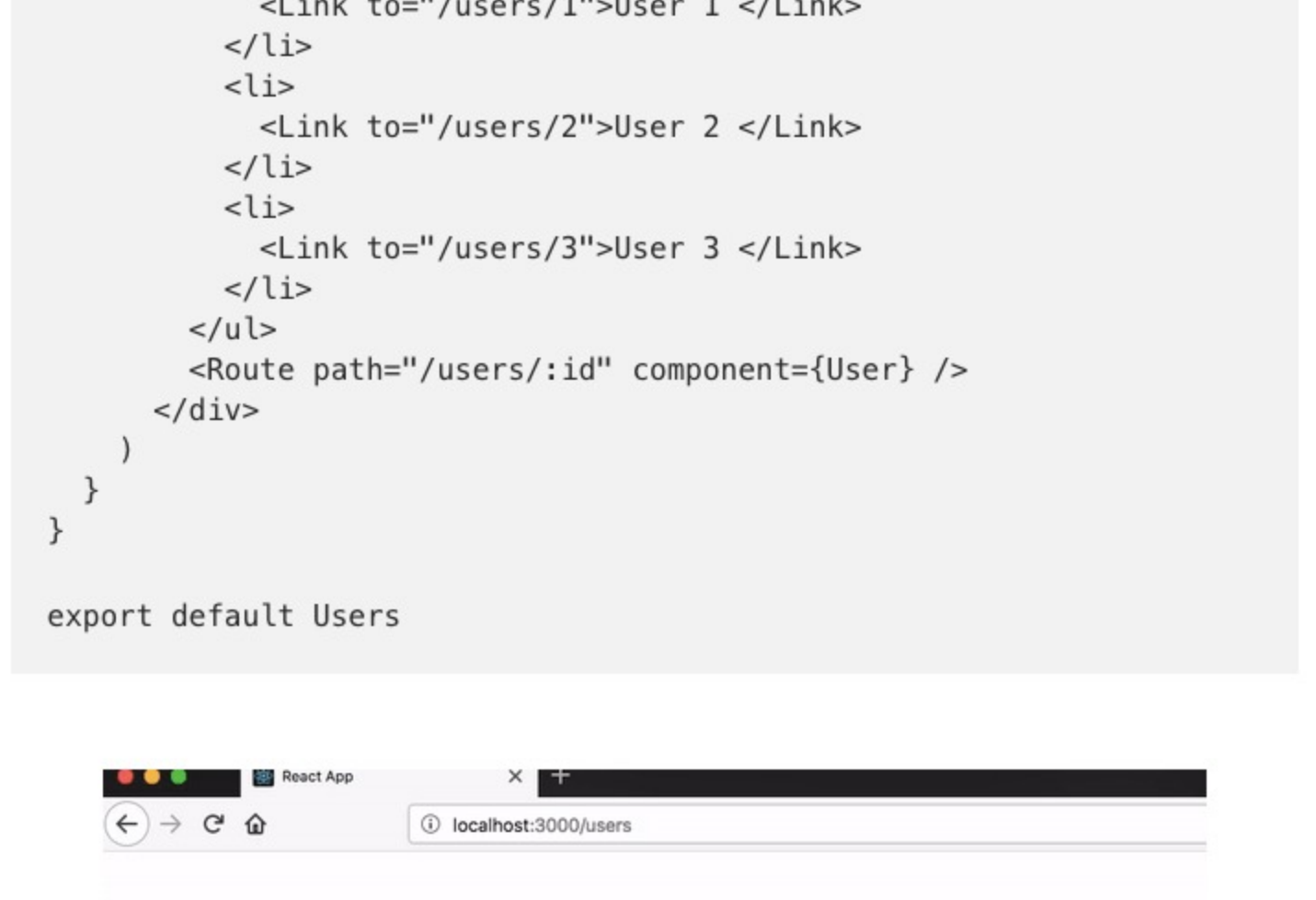
Let's render that id 1 in the screen.

users.js

```
import React from 'react'

class Users extends React.Component {
  render() {
    const { params } = this.props.match
    return (
      <div>
        <h1>Users</h1>
        <p>{params.id}</p>
      </div>
    )
  }
}

export default Users
```



Have you seen now 1 is rendered on the screen?

Nested Routes

Nested routing helps us to render the sub-routes like users/1, users/2 etc.

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>
      <hr />
      <Switch>
        <Route exact path="/" component={App} />
        <Route exact path="/users" component={Users} />
        <Route path="/contact" component={Contact} />
        <Route component={NotFound} />
      </Switch>
    </div>
  </Router>
)

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

How to implement a nested routing?

In the users.js file, we need to import the react router components because we need to implement the subroutes inside the Users Component.

users.js

```
import React from 'react'
import { Route, Link } from 'react-router-dom'

const User = ({ match }) => <p>{match.params.id}</p>

class Users extends React.Component {
  render() {
    const { url } = this.props.match
    return (
      <div>
        <h1>Users</h1>
        <strong>select a user</strong>
        <ul>
          <li>
            <Link to="/users/1">User 1</Link>
          </li>
          <li>
            <Link to="/users/2">User 2</Link>
          </li>
          <li>
            <Link to="/users/3">User 3</Link>
          </li>
        </ul>
        <Route path="/users/:id" component={User} />
      </div>
    )
  }
}

export default Users
```



active styles using NavLink

NavLink

It is used to style the active routes so that user knows on which page he or she is currently browsing on the website.

What is the difference between NavLink and Link?

The link is used to navigate the different routes on the site. But NavLink is used to add the style attributes to the active routes.

In our routing app, we have three routes which are [home,/users,/contact] Let's style them using NavLink.

We need to add a new prop called activeClassName to the NavLink component so that it applies that class whenever the route it is active.

add these styles in the index.css file

index.css

```
body {
  margin: 0;
  padding: 2rem;
  font-family: sans-serif;
}

.active{
  color:red;
}
```

index.js

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'
import {
  Route,
  NavLink,
  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>
          <NavLink exact activeClassName="active" to="/">
            Home
          </NavLink>
        </li>
        <li>
          <NavLink activeClassName="active" to="/users">
            Users
          </NavLink>
        </li>
        <li>
          <NavLink activeClassName="active" to="/contact">
            Contact
          </NavLink>
        </li>
      </ul>
      <hr />
      <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'))
```

Now you can see a red color is applied to the active routes.

Programmatically navigate

What is Programmatic navigation?

It means we need to redirect the user when an event happens on that route.

For example, when a user is successfully logged in he or she will be redirected to the home page.

How to Navigate Programmatically in react router?

To navigate programmatically we need to take the help of history object which is passed by the react router.

Let's add a contact form to our Contact component.

There is a push method available in the history object by using the push method we are redirecting the user to the Home page whenever a user submits the form.

contact.js

```
import React from 'react'

class Contact extends React.Component {
  onSubmit = () => {
    this.props.history.push('/')
  }

  render() {
    return (
      <form>
        <input placeholder="name" type="name" />
        <input placeholder="email" type="email" />
        <button onClick={this.onSubmit}>Submit</button>
      </form>
    )
  }
}

export default Contact
```

Code repository

Other interesting articles

1. [React Redux tutorial with examples](#)
2. [Render props in react](#)
3. [Animations in React](#)

originally published at [reactgo.com](#)

codeburst.io

Subscribe to CodeBurst's once-weekly **Email Blast**, Follow CodeBurst on **Twitter**, view **The 2018 Web Developer Roadmap**, and **Learn Full Stack Web Development**.

ReactReact RouterTechnologyJavaScriptReactjs

473 claps

Sai gowtham

JavaScript developer, writer my tutorials on reactgo.com

Follow

codeburst

Bursts of code to power through your day.Web Development articles, tutorials, and news.

Follow

Redux + React Router

codeburst/React-Router Private alternative

More from codeburst

Large Offline Datasets with Apollo Client

More from codeburst

Progressive Web Application Progressive State of the W

D'Ascoli

4 · 3 min read

111

🔒

John Tucker

Jan 26 · 3 min read

548

🔒

Tarique Ejaz

Jan 22 · 8 min read

🔒

Responses

Write a response...

Applause from Sai gowtham (author)

Aftab Alam

Jan 30

You have done great job. I already know how to use React Router and I often forget these details. I found your article excellent on the topic of react router.

4

🔒

Applause from Sai gowtham (author)

Ali Hassan Raza

Jan 22

It was very helpful thanks. :)

2

🔒

Conversation with Sai gowtham.

Tahsin Safa Elmali

Sep 22, 2018

index.js

Thank you for sharing! But I need to add that you missed to add the css code to the instruction. Please also add the index.css file's codes to the article.

index.css

.active{

color:red;

}

1 response

Sai gowtham

Sep 22, 2018

Sure I will add it.

🔒

Conversation with Sai gowtham.

Bingna Li

Oct 26, 2018

i have some difficulties running this tutorial but managed to fix it with help by my mentor. Here is my sharing.

<Route> should only use in the index.js file.

so it should be

<Route exact path="/users" component={Users} />

<Route path="/users/:id" component={User} />

1 response

Sai gowtham

Oct 26, 2018

There is a functional components present inside the users.js file. please check out [code repository](#)

If you have difficulties in reading long tutorial check out [my site](#) where i break down into parts.

🔒

Conversation with Sai gowtham.

Frieder Vibrans

Sep 19, 2018

Thanks for sharing :)

1 response

Sai gowtham

Sep 19, 2018

you're welcome :)

🔒

Show all responses