

REACTJS – MODULE 6.2

ROUTING IN REACT USING CHILD ROUTES AND PASSING PARAMETER



AGENDA

- Understand how child routes are created
- Understand How to set route paths for child routes
- How to pass parameter in routing
- Demo
- Activity



Routing with child routes

In the previous module we have discussed how to achieve routing in react using react-router-dom library

In this module we will focus on routing with child routes and also we will see how to pass parameters

Routing with child routes (Cont...)

Nesting components is a technique where one component will be placed inside another component

Routing in react provides a way to navigate nested components by routing with child routes and also keeping the application as single page application

As discussed in the Module 6.1 routing in react uses 3 main functionalities `<BrowserRouter>`, `<Route>` and `<Link>`

In this module we will use react-router-dom to utilize router features in the web application

Demo – Routing with child routes

Create new react project
`create-react-app routingchild`

Install react library
`npm install react-router-dom`

Demo – Routing with child routes (Cont...)

Open the project using Webstorm and create new Home Component(Home.js)

```
import React, {Component} from 'react';
class Home extends Component{
  render() {
    return (
      <div style={{ 'textAlign': 'center', 'width': '100%',
        'color': 'blue' }}>
        <h2> This is Home Component </h2></div>
      )
    }
}
export default Home;
```

Demo – Routing with child routes (Cont...)

Create another component About Component(About.js)

```
import React, {Component} from 'react';
import NavigationSection from './NavigationSection';
class About extends Component{
  render() {
    return(
      <div style={{'textAlign': 'center',
        'width': '100%', 'color': 'blue',
        'backgroundColor': 'pink'}}>
        <h2>This is about page</h2>
        <div>
          <NavigationSection/>
        </div>
      </div>
    )
  }
}
export default About;
```

Demo – Routing with child routes (Cont...)

Create another component NewIT Component(NewIT.js)

```
import React, {Component} from 'react';
class NewIT extends Component {
  render() {
    return (
      <div style={{ 'textAlign': 'center',
                    'width': '100%', 'color': 'blue',
                    'height': '80%' }}>
        <h2>NewIT Learning Path</h2>
      </div>
    )
  }
}
export default NewIT;
```


Demo – Routing with child routes (Cont...)

Create another component Digital Component(Digital.js)

```
import React, {Component} from 'react';
import {Link} from 'react-router-dom';
class Digital extends Component{
  render() {
    return(
      <div style={{ 'textAlign': 'center', 'width': '100%',
        'color': 'blue', 'height': '80%' }}>
        <h2>Digital Learning path</h2>
        <ul style={{ 'listStyle': 'none',
          'textAlign': 'center' }}>
          <li><Link to="/About/Digital/UI Path">UI Path</Link></li> <br/>
          <li><Link to="/About/Digital/Bigdata">Bigdata</Link></li>
        </ul>
      </div>
    )
  }
}
export default Digital;
```

Demo – Routing with child routes (Cont...)

Create another component NavigationSection
Component(NavigationSection.js)

```
import React, {Component} from 'react';
import {Link} from 'react-router-dom';
class NavigationSection extends Component{
  render() {
    return (
      <div style={{'textAlign': 'center',
        'width': '100%', 'color': 'blue',
        'backgroundColor': 'pink'}}>
        <Link to="/About/NewIT">NewIT</Link><span> | </span>
        <Link to="/About/Digital">Digital</Link>
      </div>
    )
  }
}
export default NavigationSection;
```

Demo – Routing with child routes (Cont...)

Inside index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import NewIT from './NewIT';
import Digital from './Digital';
import registerServiceWorker from './registerServiceWorker';
import { BrowserRouter as Router, Route } from 'react-router-dom';
import About from './About';
import Home from './Home';
import Digitalparam from './Digitalparam';
ReactDOM.render(
  <Router>
    <div>
      <Route path="/" component={App} />
      <Route path="/Home" component={Home} />
      <Route path="/About" component={About} />
      <Route path="/About/NewIT" component={NewIT} />
      <Route path="/About/Digital" component={Digital} />
    </div>
  </Router>
  , document.getElementById('root'));
registerServiceWorker();
```

Demo – Routing with child routes (Cont...)

Inside App.js

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

class App extends Component {
  render() {
    return (
      <div style={{ 'textAlign': 'center', 'width': '100%', 'color': 'blue',
        'backgroundColor': 'lightgreen' }}>
        <h2>Welcome to Routing with childs example</h2>
        <Link to="/Home">Home</Link> <span> | </span>
        <Link to="/About">About</Link>
      </div>
    );
  }
}

export default App;
```

Demo – Routing with child routes (Cont...) – Explanation

In the example initially we created 2 custom components(Home and About)

```
import React, {Component} from 'react';
class Home extends Component{
  render(){
    return(
      <div style={{'textAlign': 'center', 'width': '100%',
        'color': 'blue'}}>
        <h2> This is Home Component</h2></div>
    )
  }
}
export default Home;
```

```
import React, {Component} from 'react';
import NavigationSection from './NavigationSection';
class About extends Component{
  render() {
    return(
      <div style={{'textAlign': 'center',
        'width': '100%', 'color': 'blue',
        'backgroundColor': 'pink'}}>
        <h2>This is about page</h2>
        <div>
          <NavigationSection/>
        </div>
      </div>
    )
  }
}
export default About;
```

Demo – Routing with child routes (Cont...) – Explanation

Route information are provided inside index.js corresponding to App, Home and About components

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import NewIT from './NewIT';
import Digital from './Digital';
import registerServiceWorker from './registerServiceWorker';
import { BrowserRouter as Router, Route } from 'react-router-dom';
import About from './About';
import Home from './Home';
import Digitalparam from './Digitalparam';
ReactDOM.render(
  <Router>
    <div>
      <Route path="/" component={App} />
      <Route path="/Home" component={Home} />
      <Route path="/About" component={About} />
      <Route path="/About/NewIT" component={NewIT} />
      <Route path="/About/Digital" component={Digital} />
    </div>
  </Router>
  , document.getElementById('root'));
registerServiceWorker();
```

If the path for App component is mentioned as `<Route exact path="/">` Then the app component will display only if the url is "localhost:3000". In other cases it will not display

Demo – Routing with child routes (Cont...) – Explanation

Route information are provided inside index.js corresponding to App, Home and About components

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import NewIT from './NewIT';
import Digital from './Digital';
import registerServiceWorker from './registerServiceWorker';
import { BrowserRouter as Router, Route } from 'react-router-dom';
import About from './About';
import Home from './Home';
import Digitalparam from './Digitalparam';
ReactDOM.render(
  <Router>
    <div>
      <Route path="/" component={App} />
      <Route path="/Home" component={Home} />
      <Route path="/About" component={About} />
      <Route path="/About/NewIT" component={NewIT} />
      <Route path="/About/Digital" component={Digital} />
    </div>
  </Router>
  , document.getElementById('root'));
registerServiceWorker();
```

Route details are provided for App, Home and About component

Demo – Routing with child routes (Cont...) – Explanation

Links are provided in App.js , which are seen by the user.
And user can click on any one of the link

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

class App extends Component {
  render() {
    return (
      <div style={{ 'textAlign': 'center', 'width': '100%', 'color': 'blue',
        'backgroundColor': 'lightgreen' }}>
        <h2>Welcome to Routing with childs example</h2>
        <Link to="/Home">Home</Link> <span> | </span>
        <Link to="/About">About</Link>
      </div>
    );
  }
}
export default App;
```

These links are displayed on the view, when the user clicks on any of the link respective component is displayed

Demo – Routing with child routes (Cont...) – Explanation

Further we created 2 more custom components
NewIT and Digital

```
import React, {Component} from 'react';
class NewIT extends Component{
  render() {
    return (
      <div style={{ 'textAlign': 'center',
        'width': '100%', 'color': 'blue',
        'height': '80%' }}>
        <h2>NewIT Learning Path</h2>
      </div>
    )
  }
}
export default NewIT;
```

```
import React, {Component} from 'react';
import {Link} from 'react-router-dom';
class Digital extends Component{
  render() {
    return (
      <div style={{ 'textAlign': 'center', 'width': '100%',
        'color': 'blue', 'height': '80%' }}>
        <h2>Digital Learning path</h2>
        <ul style={{ 'listStyle': 'none',
          'textAlign': 'center' }}>
          <li><Link to="/About/Digital/UI Path">UI Path</Link></li> <br/>
          <li><Link to="/About/Digital/Bigdata">Bigdata</Link></li>
        </ul>
      </div>
    )
  }
}
export default Digital;
```

Demo – Routing with child routes (Cont...) – Explanation

Created NavigationSection component which will provide links for NewIT and Digital components. These will be childs for About component

```
import React, {Component} from 'react';
import {Link} from 'react-router-dom';
class NavigationSection extends Component{
  render() {
    return(
      <div style={{ 'textAlign': 'center',
        'width': '100%', 'color': 'blue',
        'backgroundColor': 'pink' }}>
        <Link to="/About/NewIT">NewIT</Link><span> | </span>
        <Link to="/About/Digital">Digital</Link>
      </div>
    )
  }
}
export default NavigationSection;
```

As NewIT and Digital will be childs for About component

Demo – Routing with child routes (Cont...) – Explanation

The NavigationSection component is rendered inside About component

```
import React, {Component} from 'react';
import NavigationSection from '../NavigationSection';
class About extends Component{
  render() {
    return (
      <div style={{'textAlign': 'center',
        'width': '100%', 'color': 'blue',
        'backgroundColor': 'pink'}}>
        <h2>This is about page</h2>
        <div>
          <NavigationSection/>
        </div>
      </div>
    )
  }
}
export default About;
```

The Links are displayed as part of About component

Demo – Routing with child routes (Cont...) – Explanation

Route information are provided inside index.js corresponding to NewIT and Digital child components

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import NewIT from './NewIT';
import Digital from './Digital';
import registerServiceWorker from './registerServiceWorker';
import { BrowserRouter as Router, Route } from 'react-router-dom';
import About from './About';
import Home from './Home';
import Digitalparam from './Digitalparam';
ReactDOM.render(
  <Router>
    <div>
      <Route path="/" component={App}/>
      <Route path="/Home" component={Home}/>
      <Route path="/About" component={About}/>
      <Route path="/About/NewIT" component={NewIT}/>
      <Route path="/About/Digital" component={Digital}/>
    </div>
  </Router>
  , document.getElementById('root'));
registerServiceWorker();
```

About acts like parent component and NewIT and Digital are childs for About components. Hence the path is referred using About component

Demo – Routing with child routes (Cont...)

Output



Welcome to Routing with childs example

[Home](#) | [About](#)

Demo – Routing with child routes (Cont...)

Output when clicked
Home Component



Welcome to Routing with childs example

[Home](#) | [About](#)

This is Home Component

Demo – Routing with child routes (Cont...)

Output when clicked
About Component



Welcome to Routing with childs example

[Home](#) | [About](#)

This is about page

[NewIT](#) | [Digital](#)

Demo – Routing with child routes (Cont...)

Output when clicked
NewIT Component



Welcome to Routing with childs example

[Home](#) | [About](#)

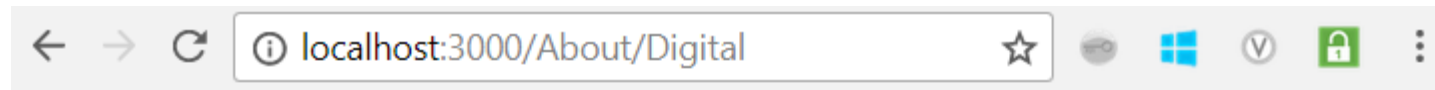
This is about page

[NewIT](#) | [Digital](#)

NewIT Learning Path

Demo – Routing with child routes (Cont...)

Output when clicked
Digital Component



Welcome to Routing with childs example

[Home](#) | [About](#)

This is about page

[NewIT](#) | [Digital](#)

Digital Learning path

[UI Path](#)

[Bigdata](#)

Activity

1. Create a new component named “Contact” component
2. Create two more components named “Email” and “Feedback” components
3. Email and Feedback components must be child for Contact component
4. Apply child routing concept and complete the activity

Routing with passing parameter

Routing in react provides the feature of passing the data from one component to other component

The data which we need to pass to other component it is referred as parameter

Demo – Routing by passing parameter

Use existing Digital Component(Digital.js)

```
import React, {Component} from 'react';
import {Link} from 'react-router-dom';
class Digital extends Component{
  render() {
    return(
      <div style={{ 'textAlign': 'center', 'width': '100%',
        'color': 'blue', 'height': '80%' }}>
        <h2>Digital Learning path</h2>
        <ul style={{ 'listStyle': 'none',
          'textAlign': 'center' }}>
          <li><Link to="/About/Digital/UI Path">UI Path</Link></li> <br/>
          <li><Link to="/About/Digital/Bigdata">Bigdata</Link></li>
        </ul>
      </div>
    )
  }
}
export default Digital;
```

Demo – Routing by passing parameter (Cont...)

Create new component
Digitalparam(Digitalparam.js)

```
import React, {Component} from 'react';
class Digitalparam extends Component
{
    render() {
        return (
            <div style={{ 'textAlign': 'center' }}>
                <h3>You have selected
                    {this.props.match.params.id} course</h3>
            </div>
        )
    }
}
export default Digitalparam;
```

Demo – Routing by passing parameter (Cont...)

Inside index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import NewIT from './NewIT';
import Digital from './Digital';
import registerServiceWorker from './registerServiceWorker';
import { BrowserRouter as Router, Route } from 'react-router-dom';
import About from './About';
import Home from './Home';
import Digitalparam from './Digitalparam';
ReactDOM.render(
  <Router>
    <div>
      <Route path="/" component={App} />
      <Route path="/Home" component={Home} />
      <Route path="/About" component={About} />
      <Route path="/About/NewIT" component={NewIT} />
      <Route path="/About/Digital" component={Digital} />
      <Route path="/About/Digital/:id" component={Digitalparam} />
    </div>
  </Router>

  , document.getElementById('root'));
registerServiceWorker();
```

Demo – Routing by passing parameter (Cont...) – Explanation

Digital component have 2 Links “UI Path” and “Bigdata”. These will be seen by the user and user can click on any one of these. When this is clicked the corresponding data is captured and sent to other component(Digitaldetail)

```
import React, {Component} from 'react';
import {Link} from 'react-router-dom';
class Digital extends Component{
  render(){
    return(
      <div style={{'textAlign':'center','width':'100%',
        'color':'blue','height':'80%'}}>
        <h2>Digital Learning path</h2>
        <ul style={{'listStyle':'none',
          'textAlign':'center'}}>
          <li><Link to="/About/Digital/UI Path">UI Path</Link></li> <br/>
          <li><Link to="/About/Digital/Bigdata">Bigdata</Link></li>
        </ul>
      </div>
    )
  }
}
export default Digital;
```

When the user clicks on the link then corresponding parameter will be sent

Demo – Routing by passing parameter (Cont...)

Inside index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import NewIT from './NewIT';
import Digital from './Digital';
import registerServiceWorker from './registerServiceWorker';
import { BrowserRouter as Router, Route } from 'react-router-dom';
import About from './About';
import Home from './Home';
import Digitalparam from './Digitalparam';

ReactDOM.render(
  <Router>
    <div>
      <Route path="/" component={App} />
      <Route path="/Home" component={Home} />
      <Route path="/About" component={About} />
      <Route path="/About/NewIT" component={NewIT} />
      <Route path="/About/Digital" component={Digital} />
      <Route path="/About/Digital/:id" component={Digitalparam} />
    </div>
  </Router>

  , document.getElementById('root'));
registerServiceWorker();
```

When the Digital component is displayed it will have links for UI Path and Bigdata, and when user clicks on any one link the respective parameter is passed with the name of id

Also when id is passed as a parameter, in this situation Digitalparam component will be displayed

Demo – Routing by passing parameter (Cont...)

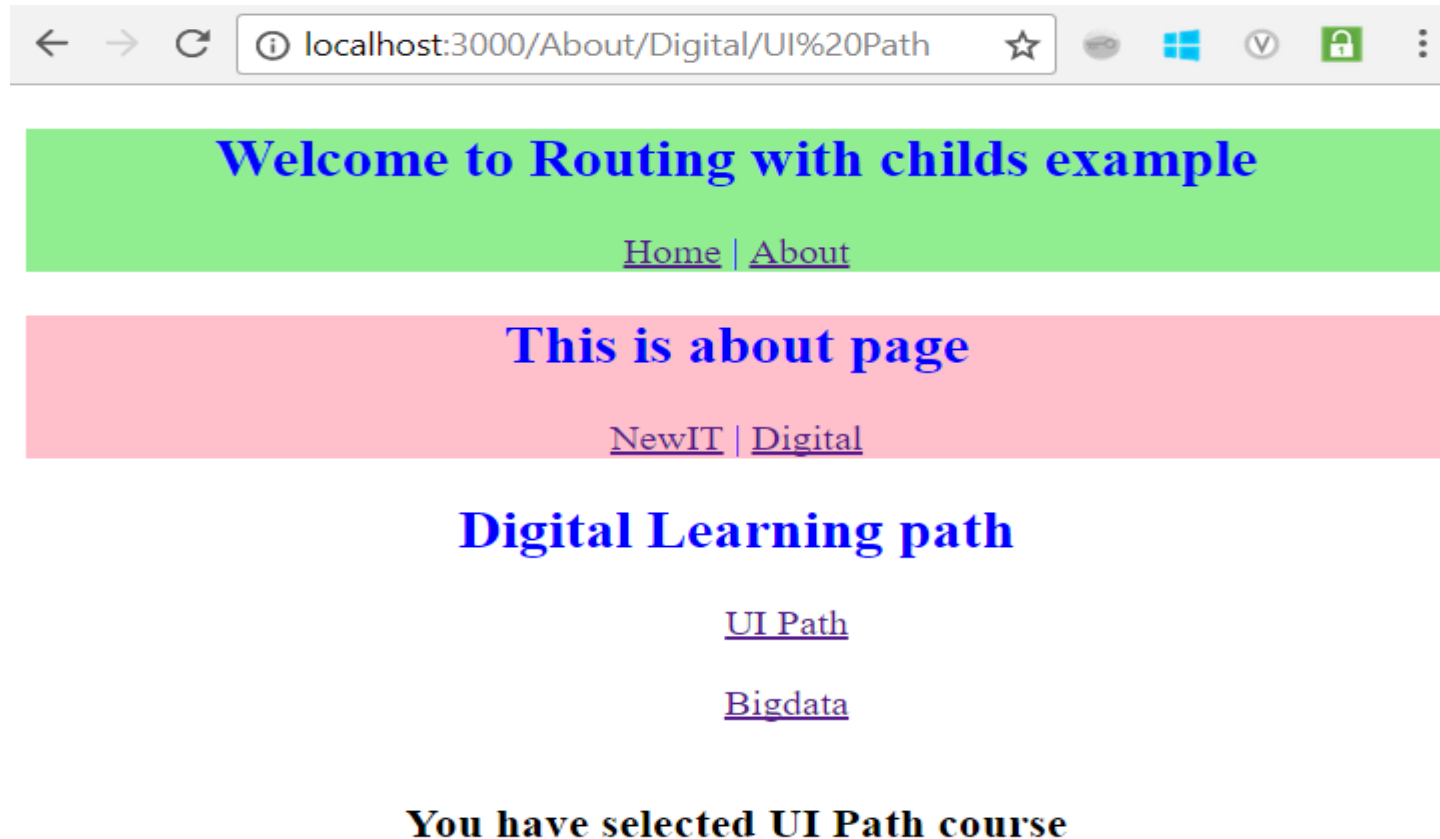
Digitalparam component will display parameter coming from Digital component

```
import React, {Component} from 'react';
class Digitalparam extends Component
{
  render() {
    return (
      <div style={{'textAlign': 'center'}}>
        <h3>You have selected
          {this.props.match.params.id} course</h3>
        </div>
      )
    }
}
export default Digitalparam;
```

Displays which parameter is passed from Digital component

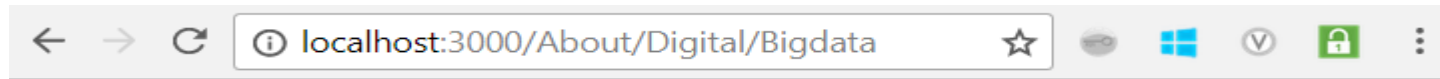
Demo – Routing by passing parameter (Cont...)

Output when clicked UI
Path link



Demo – Routing by passing parameter (Cont...)

Output when clicked
Bigdata link



Welcome to Routing with childs example

[Home](#) | [About](#)

This is about page

[NewIT](#) | [Digital](#)

Digital Learning path

[UI Path](#)

[Bigdata](#)

You have selected Bigdata course

Activity

1. Use existing Digital and Digitalparam components and perform following
2. When the user clicks on “UI Path” the below view should be displayed



3. When the user clicks on “Bigdata” then the below view should be displayed



MODULE SUMMARY

- How to create child routes
- How to configure route path
- How parameters are passed
- How to use match
- How to receive parameter using match



THANK YOU

