Note: The below code is not final

index.js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import 'bootstrap/dist/css/bootstrap.min.css'

import \* as serviceWorker from './serviceWorker';

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

// If you want your app to work offline and load faster, you can change

// unregister() to register() below. Note this comes with some pitfalls.

// Learn more about service workers: https://bit.ly/CRA-PWA

serviceWorker.unregister();

App.js

import React, {Component} from 'react';

import {Route, BrowserRouter, Link, Switch} from 'react-router-dom';

class RegistrationForm extends Component {

constructor(props){

super(props);

this.handleChange = this.handleChange.bind(this);

this.handleSubmit = this.handleSubmit.bind(this);

this.state = {id:'', name : '', salary : ''}

}

handleChange(event) {

let tempName = event.target.name;

let tempValue = event.target.value;

this.setState({[tempName]: tempValue});

}

handleSubmit(event) {

console.log(this.state);

event.preventDefault();

}

render() {

return(<div>

<h3>Registration Form</h3>

<form onSubmit = {this.handleSubmit}>

Id : <input type = "text" name = "id" onChange = {this.handleChange}></input>

<br/>

Name : <input type = "text" name = "name" onChange = {this.handleChange}></input>

<br/>

Salary : <input type = "text" name = "salary" onChange = {this.handleChange}></input>

<br/>

<input type = "submit" value = "Submit"></input>

</form>

<Link to = "/login">Goto Login</Link>

</div>)

}

}

// Login Form

class LoginForm extends Component {

constructor(props){

super(props);

this.handleChange = this.handleChange.bind(this);

this.handleSubmit = this.handleSubmit.bind(this);

this.state = {id:'', name : ''}

}

handleChange(event) {

let tempName = event.target.name;

let tempValue = event.target.value;

this.setState({[tempName]: tempValue});

}

handleSubmit(event) {

event.preventDefault();

this.props.history.push(`/success/${this.state.id}`);

}

render() {

return(<div>

<h3>Login Form!</h3>

<form onSubmit = {this.handleSubmit}>

Id : <input type = "text" name = "id" onChange = {this.handleChange}></input>

<br/>

Name : <input type = "text" name = "name" onChange = {this.handleChange}></input>

<br/>

<input type = "submit" value = "Submit"></input>

</form>

<Link to = "/register">Register</Link>

</div>)

}

}

class Success extends Component {

componentDidMount() {

console.log('---component did mount---');

console.log(this.props.match.params);

console.log('---component did mount---');

}

render() {

//console.log(this.props.match.url);

return (<div>

<div className = "row">

<div className = "col-3">

<h6 className = "text-center">Profile Summary</h6>

<hr />

<p>Id: {this.props.match.params.id}</p>

</div>

<div className = "col-1">

</div>

<div className = "col-8">

<h6>This is success component : {this.props.match.params.id}</h6>

<hr/>

<Link to = {this.props.match.url+'/dashboard'}>Dashboard</Link> /

<Link to = {this.props.match.url+'/addContacts'}>Add Contact</Link> /

<Link to = {this.props.match.url+'/showAll'}>Show Contacts</Link> /

<Link to = {this.props.match.url+'/update'}>Update</Link>

<Route exact path = {this.props.match.path} component = {Dashboard}></Route>

<Route path = {this.props.match.path+"/dashboard"} component = {Dashboard}></Route>

<Route path = {this.props.match.path+"/addContacts"} component = {AddContacts}></Route>

<Route path = {this.props.match.path+"/showAll"} component = {ShowContacts}></Route>

<Route path = {this.props.match.path+"/update"} component = {UpdateProfile}></Route>

</div>

</div>

</div>);

}

}

class Dashboard extends Component {

render() {

return(<div>

<h6>Dashboard of {this.props.match.params.id}</h6>

</div>)

}

}

class AddContacts extends Component {

render() {

return(<div>

<h6>Add Contacts of {this.props.match.params.id}</h6>

</div>)

}

}

class ShowContacts extends Component {

render() {

return(<div>

<h6>Show All Contacts of {this.props.match.params.id}</h6>

</div>)

}

}

class UpdateProfile extends Component {

render() {

console.log('update profile', this.props.match.params);

return(<div>

<h6>Update Profile of {this.props.match.params.id}</h6>

</div>)

}

}

class App extends Component {

render() {

return (

<div className = "container-fluid">

<h1 className = "text-center">Contact Hub in React</h1> <hr />

<BrowserRouter>

<Switch>

<Route exact path = "/" component = {RegistrationForm}></Route>

<Route path = "/register" component = {RegistrationForm}></Route>

<Route path = "/login" component = {LoginForm}></Route>

<Route path = "/success/:id" component = {Success}></Route>

</Switch>

</BrowserRouter>

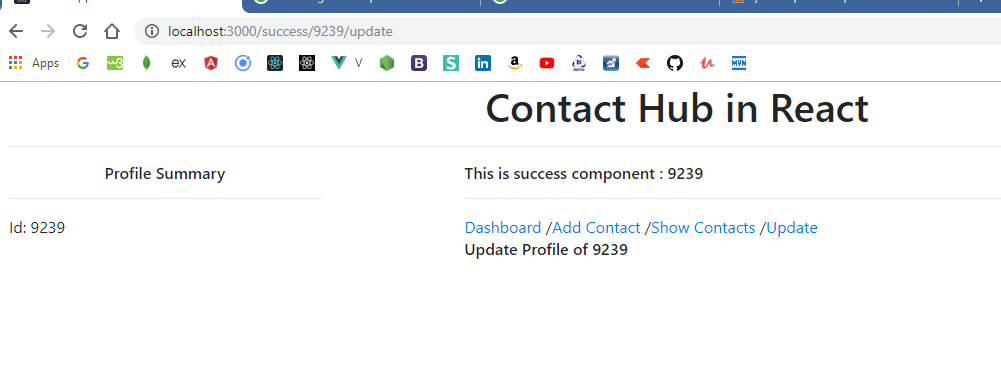
</div>

);

}

}

export default App;

Partial Output  


<Link> and <a>:

Link and anchor both produces the url but Link gives instruction to the React BrowserRouter component to add the component to the Route that matches to the URL without refreshing the whole document, whereas the anchor gives instruction the Browser directly that will load the whole page instead of updating the DOM.

Things to understand is <Link> is react component that updates the DOM and <a> updates the whole Document.

Route: It is like a outlet for the components that has to be rendered when a path matches.

Below is the code with some working UI of contact manager but Dashboard doesn’t show newly added counts it must be modified, but if you refresh the whole document it will show the newly added records count.

App.js

import React, { Component } from 'react';

import './App.css';

import {Route, BrowserRouter, Link, Switch} from 'react-router-dom';

import axios from 'axios';

class DashBoardComponent extends Component {

render() {

return(<div>

<h4>Hi, {this.props.profile.name}</h4> <hr />

<h6>Your total contacts: {this.props.profile.contactList.length}</h6>

</div>);

}

}

class AddContactComponent extends Component {

constructor(props) {

super(props);

this.state = {name : '', phone : '', message : ''};

}

onSubmit = (e) => {

e.preventDefault();

let userId = this.props.profile.userId;

let addContactURL = `http://localhost:8080/my-contact-app/api/profile/${userId}/addContact`;

axios.post(addContactURL, this.state).then((response) => {

this.setState({message : response.data.message})

});

}

onChange = (e) => {

let tempName = e.target.name;

let tempValue = e.target.value;

this.setState({

[tempName] : tempValue

});

}

render() {

return(<div>

<h4>Add Contacts</h4>

<hr />

<form onSubmit = {this.onSubmit}>

<input type = "text" name = "name" placeholder = "Enter Name" onChange = {this.onChange}></input> <br />

<input type = "number" name = "phone" placeholder = "Enter Phone No" onChange = {this.onChange}></input> <br />

<input type = "submit" value = "Add"></input>

</form>

<h6>{this.state.message}</h6>

</div>);

}

}

class ShowAllContacts extends Component {

render() {

let contactsList = this.props.profile.contactList;

return(<div>

<h4>All Contacts</h4>

<table className = "table table-striped">

<thead>

<tr>

<th>Sl No</th><th>Name</th><th>Phone Number</th>

</tr>

</thead>

<tbody>

{contactsList.map((contact, index) =>

<tr key = {index}>

<td>{index + 1}</td><td>{contact.name}</td><td>{contact.phone}</td>

</tr>)}

</tbody>

</table>

</div>);

}

}

class UpdateProfile extends Component {

render() {

return(<div>

<h5>Update Profile </h5>

</div>);

}

}

class ProfileComponent extends Component {

constructor(props) {

super(props);

this.state = {profile : undefined}

}

componentDidMount() {

let userId = this.props.match.params.id;

let fetchOneUrl = `http://localhost:8080/my-contact-app/api/profile/${userId}`;

axios.get(fetchOneUrl).then(response => {

this.setState({profile : response.data});

});

}

render() {

if(this.state.profile === undefined)

return(<div><h5>Loading...</h5></div>);

return(

<div className = "container-fluid">

<div className = "row">

<div className = "col-3">

<h5>Profile Summary</h5> <hr />

<table className = "table table-borderless">

<thead>

<tr>

<th>Birthday: </th><td>{this.state.profile.dob}</td>

</tr>

<tr>

<th>Mobile No.: </th><td>{this.state.profile.phone}</td>

</tr>

</thead>

</table>

</div>

<div className = "col-1">

<div style = {{height : '500px', borderLeft : '1px solid silver'}}></div>

</div>

<div className = "col-8">

<nav className = "navbar nav-default navbar-expand-sm">

<ul className = "nav navbar-nav ml-auto">

<li>

<Link to = {this.props.match.url+'/dashboard'} className = "mr-3">Dashboard</Link>

</li>

<li>

<Link to = {this.props.match.url+'/addContact'} className = "mr-3">Add Contact</Link>

</li>

<li>

<Link to = {this.props.match.url+'/showAll'} className = "mr-3">Show My Contacts</Link>

</li>

<li>

<Link to = {this.props.match.url+'/updateProfile'} className = "mr-3">Update</Link>

</li>

<li>

<Link to = "/login" className = "mr-3">Logout</Link>

</li>

</ul>

</nav>

<Route exact path = {this.props.match.path} component = {() => <DashBoardComponent profile = {this.state.profile}/>}></Route>

<Route path = {this.props.match.path+"/dashboard"} component = {() => <DashBoardComponent profile = {this.state.profile}/>}></Route>

<Route path = {this.props.match.path+"/addContact"} component = {() => <AddContactComponent profile = {this.state.profile}/>}></Route>

<Route path = {this.props.match.path+"/showAll"} component = {() => <ShowAllContacts profile = {this.state.profile}/>}></Route>

<Route path = {this.props.match.path+'/updateProfile'} component = {() => <UpdateProfile profile = {this.state.profile}/>}></Route>

<Route exact path = "/login" component = {LoginComponent}></Route>

</div>

</div>

</div>

)

}

}

class RegisterComponent extends Component {

constructor(props) {

super(props);

this.state = {name : "", dob : "", password : "", phone : ""};

}

handleChange = (e) => {

let inputName = e.target.name;

let inputValue = e.target.value;

this.setState({[inputName] : inputValue});

}

handleSubmit = (e) => {

e.preventDefault();

let registerURL = "http://localhost:8080/my-contact-app/api/createProfile";

axios.post(registerURL, this.state)

.then(response => {

if(response.status) {

//this.props.history.push(`login/${response.data.userId}`);

alert(`Your profile is registered with an User Id: ${response.data.userId}`)

this.props.history.push(`login`);

}

});

}

render() {

return(<div>

<h3>Register</h3>

<hr />

<form onSubmit = {this.handleSubmit}>

<input type = "text" name = "name" placeholder = "Enter Name" onChange = {this.handleChange}></input>

<input type = "password" name = "password" placeholder = "Enter Password" onChange = {this.handleChange}></input>

<br />

Birthday : <input type = "date" name = "dob" onChange = {this.handleChange} />

<br />

<input type = "text" name = "phone" placeholder = "Enter Mobile No" onChange = {this.handleChange}/>

<br />

<input type = "submit" value = "Sign Up" />

</form>

<Link to = "login">Sign In</Link>

</div>);

}

}

class LoginComponent extends Component {

constructor(props) {

super(props);

this.state = {id : "", password : ""};

}

handleChange = (e) => {

let inputName = e.target.name;

let inputValue = e.target.value;

this.setState({[inputName] : inputValue});

}

handleSubmit = (e) => {

e.preventDefault();

let loginURL =

`http://localhost:8080/my-contact-app/api/login/${this.state.id}/${this.state.password}`;

axios.get(loginURL).then((response) => {

if(response.status === 200) {

this.props.history.push(`profile/${this.state.id}`);

}

}).catch(error => {alert("Wrong credentials please login again")});

}

render() {

return(<div>

<h3>Login</h3> <hr />

<form onSubmit = {this.handleSubmit}>

<input type = "text" name = "id" onChange = {this.handleChange}

placeholder = "Enter User Id"></input>

<input type = "text" name = "password" onChange = {this.handleChange}

placeholder = "Enter Password"></input>

<input type = "submit" value = "Sign In"/>

</form>

<Link to = "register">Sign Up</Link>

</div>);

}

}

class App extends Component {

render() {

return (<div className = "container-fluid">

<div className = "text-center">

<h2>Contact Manager Project</h2>

</div>

<hr />

<div className = "container-fluid">

<BrowserRouter>

<Switch>

<Route exact path = "/" component = {LoginComponent}></Route>

<Route path = "/login" component = {LoginComponent}></Route>

{/\* <Route path = "/login/:id" component = {LoginComponent}></Route> \*/}

<Route path = "/register" component = {RegisterComponent}></Route>

<Route path = "/profile/:id" component = {ProfileComponent}></Route>

</Switch>

</BrowserRouter>

</div>

</div>);

}

}

export default App;

Output:

