**Day -2 Shopping Cart**

**App.js**

import React,{Component} from "react";

//import '../node\_modules/bootstrap/dist/css/bootstrap.css';

export default class App extends Component{

render(){

return <div>

<h1 className="well"> Welcome</h1>

</div>

}

}

// We need to Use it in index.js --> export defalut

Index.js

//Steps for Component Based on Class

import App from './components/App';

import ReactDOM from "react-dom"

import React from "react"

import '../node\_modules/bootstrap/dist/css/bootstrap.css';

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

App.js – Nav Bar & AHref

import React, { Component } from "react";

export default class App extends Component {

state = { appheading: " My Shopping App" }

render() {

return <div>

<nav className="navbar navbar-inverse">

<a href="" className="navbar-brand" >{this.state.appheading} </a>

</nav>

</div>

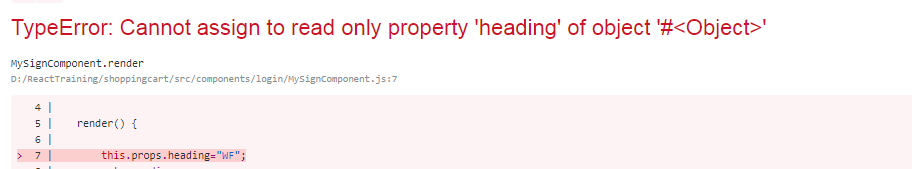
}

}

// We need to Use it in index.js --> export defalut

Difference between Props & State

PROPS – Immutable – They cannot be altered



State- Mutable

Ref Id – Document .getElemetId.value

Retrieve by “refs”

This.refs.userid.value

import React, { Component } from "react";

export default class MySignComponent extends Component {

render() {

//this.props.heading="WF";

return <div>

<h3>{this.props.heading} </h3>

<form>

<input type="text" placeholder = "UserName" ref="userid"/>

<br/>

<input type="password" placeholder = "password" ref="pwd"/>

<br/>

<button type= "button"className="btn btn-primary"

onClick={this.sendData}>{this.props.heading}</button>

<br/>

</form>

</div>

}

sendData = () =>{

console.log("Send Data Function called");

console.log(this.refs.userid.value);

}

}

SignIn.js

import React, { Component } from "react";

import MySign from "./MySignComponent";

export default class SignIn extends Component {

state = {loggedUSerName : "Guest"}

handleSignin = (uname) => {

this.setState({loggedUSerName : uname})

}

render() {

return <div>

<h1>Welcome {this.state.loggedUSerName}</h1>

<MySign heading="Sign In" onSignIn={this.handleSignin}/>

</div>

}

}

// We need to Use it in index.js --> export defalut

MyComponent.js

import React, { Component } from "react";

export default class MySignComponent extends Component {

render() {

//this.props.heading="WF";

return <div>

<h3>{this.props.heading} </h3>

<form>

<input type="text" placeholder = "UserName" ref="userid"/>

<br/>

<input type="password" placeholder = "password" ref="pwd"/>

<br/>

<button type= "button"className="btn btn-primary"

onClick={this.sendData}>{this.props.heading}</button>

<br/>

</form>

</div>

}

sendData = () =>{

console.log("Send Data Function called");

console.log(this.refs.userid.value);

this.props.onSignIn(this.refs.userid.value)

}

}

**Single Page Application**

**URL htttp://localhost:3000/<Base Path>**

<head>  
  <base href="https://www.w3schools.com/images/" target="\_blank">  
</head>

SPA 🡪 Pages

In REACT 🡪 Layout Component all- Partials are Also components.

APP.js 🡪 Layout component

Based on the path 🡪 move to specific Path for

Each partials – (Path Maps to Partials or component) instead of <anchor tag

We can use the link tag in react. React Router Library maintains the Path Mapping for Navigation od Specific components

REACT ROUTER

1. View changes in a react Application
2. Functions – Show / Hide
3. Update the URL
4. React-router , react-router-dom, react-router-native.

Install the react-router-dome :

npm I –S react-router-dom

Browser Router – for the Sync with URL

Route – Based on User component – displays the matched component – Hidden or viewable one.

Link Tag – Instead of Anchor tag use the Link Tag

ReactDOM.render(<BrowserRouter>

<App/>

</BrowserRouter>,document.getElementById("root"));

Link Example – App.js

import React, { Component } from "react";

import SignIn from "./login/SignIn";

import SignUp from "./login/SignUp";

import {Route} from "react-router-dom";

import {Link} from "react-router-dom";

export default class App extends Component {

state = { appheading: " My Shopping App" }

render() {

return <div>

<nav className="navbar navbar-inverse">

<Link to="/" className="navbar-brand" >{this.state.appheading} </Link>

<ul className= "nav navbar-nav">

<li> <Link to= "/">Home</Link></li>

<li> <Link to= "/signin">SignIn</Link></li>

<li> <Link to= "/signup">SignUp</Link></li>

</ul>

</nav>

<Route path="/" exact render= {() => <h1>Welcome Home</h1>} />

<Route path="/signin" component={SignIn} />

<Route path="/signup" component={SignUp} />

</div>

}

}

// We need to Use it in index.js --> export defalut

ROUTING with SWITHC & LINK – 400 \*

import React, { Component } from "react";

import SignIn from "./login/SignIn";

import SignUp from "./login/SignUp";

import {Route} from "react-router-dom";

import {Link,Switch} from "react-router-dom";

import {NotFound} from "./routeComponent";

export default class App extends Component {

state = { appheading: " My Shopping App" }

render() {

return <div>

<nav className="navbar navbar-inverse">

<Link to="/" className="navbar-brand" >{this.state.appheading} </Link>

<ul className= "nav navbar-nav">

<li> <Link to= "/">Home</Link></li>

<li> <Link to= "/signin">SignIn</Link></li>

<li> <Link to= "/signup">SignUp</Link></li>

</ul>

</nav>

<Switch>

<Route path="/" exact render= {() => <h1>Welcome Home</h1>} />

<Route path="/signin" component={SignIn} />

<Route path="/signup" component={SignUp} />

<Route path="\*" component={NotFound} />

</Switch>

</div>

}

}

// We need to Use it in index.js --> export defalut

RouteComponent.js

import React, { Component } from "react";

export class NotFound extends Component{

render(){

return <h1 className="well"> 404. Pg Not found !! check Links</h1>

}

}

Object Lifecycle

Init – The Object 🡪 Contructor

Setters/ Getters - Process of the oBject

Destroy the Object

Component Life Cycle.

Index.js -- > we are using a App tag 🡪 react Instantiates 🡪 rendering starts

Stages for a react component

1. Creation of a component
2. Before rendering – List of States to be listed –*ComponentWillMount Method*
3. While rendering – Display of View- *render()*
4. After rendering- *Components Did Mount*
5. Destroying a component-*Components willIUnMouint*

Mounting – Inserting a component in to the DOM 🡪 Mounting

Un Mounting 🡪removing component in to the DOM 🡪 UnMounting

Methods:

1. ComponentWillMount(); -- Before rendering
2. Render() – While Rendering
3. ComponentDidMount()-After Rendering
4. ComponentWillUnMount()- Destroying

**Let us try it in the MySignComponent**

import React, { Component } from "react";

export default class MySignComponent extends Component {

render() {

//this.props.heading="WF";

console.log("While Rendering");

return <div>

<h3>{this.props.heading} </h3>

<form>

<input type="text" placeholder = "UserName" ref="userid"/>

<br/>

<input type="password" placeholder = "password" ref="pwd"/>

<br/>

<button type= "button"className="btn btn-primary"

onClick={this.sendData}>{this.props.heading}</button>

<br/>

</form>

</div>

}

sendData = () =>{

console.log("Send Data Function called");

console.log(this.refs.userid.value);

this.props.onSignIn(this.refs.userid.value)

}

constructor(){

super();

console.log("MySign-Constructor");

}

componentWillMount(){

console.log("MySign-componentWillMount");

}

componentDidMount(){

console.log("MySign-componentDidMount");

}

componentWillUnmount(){

console.log("MySign-componentWillUnmount");

}

}

**MyComponent.js**

import React, { Component } from "react";

export default class MySignComponent extends Component {

render() {

//this.props.heading="WF";

console.log("While Rendering");

return <div>

<h3>{this.props.heading} </h3>

<form>

<input type="text" placeholder = "UserName" ref="userid"/>

<br/>

<input type="password" placeholder = "password" ref="pwd"/>

<br/>

<button type= "button"className="btn btn-primary"

onClick={this.sendData}>{this.props.heading}</button>

<br/>

</form>

</div>

}

sendData = () =>{

console.log("Send Data Function called");

console.log(this.refs.userid.value);

this.props.onSignIn(this.refs.userid.value)

}

constructor(){

super();

console.log("MySign-Constructor");

}

componentWillMount(){

console.log("MySign-componentWillMount ::"+this.props.heading);

}

componentDidMount(){

console.log("MySign-componentDidMount ::" +this.props.heading);

document.querySelector("h3").style.color="red";

}

componentWillUnmount(){

console.log("MySign-componentWillUnmount");

}

}