**==========================================================**

**https://shorturl.at/sMS17**

**Setting and Changing State**

**Props**

**Inputs**

**Lifecycle Methods**

**API Calls**

**Imports and Exports**

**Single Page Application**

**Nested Routing**

**Lazy Loading**

**==========================================================**

==========================================================

Setting and Changing State

==========================================================

import React from 'react'

export default class ChageState extends React.Component {

constructor() {

super()

this.state = {

count: 0

}

}

render() {

return (

<div className='container mt-5'>

<h1>Welcome to Change State example</h1>

<button className='btn btn-outline-primary' onClick={this.dec}> - </button>

<button className='btn btn-secondary mx-3'>{this.state.count} </button>

<button className='btn btn-outline-primary' onClick={this.inc}> + </button>

</div>

)

}

dec = () => {

this.setState({

count: this.state.count

})

this.setState(prevState => {

prevState.count -= 1

})

}

inc = () => {

this.setState({

count: this.state.count

})

this.setState(prevState => {

prevState.count += 1

})

}

}

==========================================================

props:-

==========================================================

- props are used to share data between components.

<>

src

propseg

- First.js

- Second.js

- index.js

\*\*\*First.js\*\*\*

import React from "react";

import Second from "./Second";

export default class First extends React.Component {

constructor() {

super()

this.state = {

tech: `Javascript`,

number: 100,

flag: true,

obj: {

fruit: 'Mango'

},

subs: [` Bootstrap `, ` Express `, ` React`],

products: [

{ "p\_id": 111, "p\_name": "P\_one", "p\_cost": 10000 },

{ "p\_id": 222, "p\_name": "P\_two", "p\_cost": 20000 },

{ "p\_id": 333, "p\_name": "P\_three", "p\_cost": 30000 },

{ "p\_id": 444, "p\_name": "P\_four", "p\_cost": 40000 },

{ "p\_id": 555, "p\_name": "P\_five", "p\_cost": 50000 }

]

}

}

render() {

return (

<div className="container mt-5">

<h1 className="text-primary">Welcome to props example</h1>

<p>{this.state.tech} </p>

<button className="btn btn-outline-info" onClick={this.chState}>Change State </button>

<Second data1={this.state.tech}></Second>

<Second data2={this.state.number}></Second>

<Second data3={this.state.flag}></Second>

<Second data4={this.state.obj}></Second>

<Second data5={this.state.subs}></Second>

<Second data6={this.state.products}></Second>

</div>

)

}

chState = () => {

this.setState({

tech: 'ReactJS'

})

}

}

\*\*\*Second.js\*\*\*

import React from "react";

export default class Second extends React.Component {

render() {

return (

<div>

<h2 style={{ color: 'red' }}>{this.props.data1} </h2>

<h2 style={{ color: 'green' }}>{this.props.data2} </h2>

<h2 style={{ color: 'blue' }}>{JSON.stringify(this.props.data3)} </h2>

<h2 style={{ color: 'maroon' }}>{JSON.stringify(this.props.data4)} </h2>

<h2 style={{ color: 'navy' }}>{this.props.data5} </h2>

<h2 style={{ color: 'darkgreen' }}>{JSON.stringify(this.props.data6)}</h2>

</div>

)

}

}

==========================================================

Form inputs

==========================================================

import React from 'react'

export default class FormInput extends React.Component {

constructor() {

super()

this.state = {

status: "Please Login"

}

}

render() {

return (

<div className='container mt-5'>

<form onSubmit={this.login}>

<input type='text'

placeholder='Enter Username'

name='uname' />

<br /><br />

<input type='password'

placeholder='Enter Password'

name='upwd' />

<br /><br />

<input type='submit' value='Login' />

</form>

<h1 style={{ color: 'blue' }}>{this.state.status} </h1>

</div>

)

}

login = (e) => {

e.preventDefault()

let uname = e.target.uname.value

let upwd = e.target.upwd.value

if (uname === 'admin' && upwd === 'admin')

this.setState({ status: 'Login Success' })

else

this.setState({ status: 'Login Failed' })

}

}

==============================================

Lifecycle Methods

==============================================

\*\*\*Parent.js\*\*\*

import React from "react";

import Child from "./Child";

export default class Parent extends React.Component {

constructor() {

super()

console.log('Parent Constructor called')

/\*

- constructor gets called at loading of component

- constructor will execute only once.

- it is recommended to define the state in the constructor.

\*/

this.state = {

technology: 'ReactJS'

}

}

componentWillMount() {

console.log('Parent componentWillMount')

/\*

- This method executes after constructor

- This method executes only once.

- It is recommended to change initial state in this method.

- It is recommended to set global parameters here

\*/

if (window.innerWidth < 600)

this.setState({

width: "Small Page"

})

}

render() {

console.log('Parent render')

/\*

- render is mandatory lifecycle method

- it will execute after the componentWillMount method.

- we place presentation logic here

- it will always gets called when state change

\*/

return (

<div>

<h1>Parent Component</h1>

<p style={{ color: 'blue' }}>{this.state.technology} </p>

<p style={{ color: 'red' }}>{this.state.width} </p>

<Child key1={this.state.technology} />

<button onClick={() => this.setState({ technology: 'MERN' })}>Change</button>

</div>

)

}

/\*

--- Execution Flow ---

- Parent Constructor

- Parent componentWillMount

- Parent render

- Child Constructor

- Child componentWillMount

- Child render

- if state / props change detected

- Parent render

- Child render

\*/

componentDidMount() {

console.log('Parent componentDidMount called')

}

/\*

--- Execution Flow ---

- Parent Constructor

- Parent componentWillMount

- Parent render

- Child Constructor

- Child componentWillMount

- Child render

- if state / props change detected

- Parent render

- Child render

- Child componentDidMount

- Parent componentDidMount

\*/

componentWillReceiveProps() {

/\*

- this method will execute if component receives props

\*/

console.log('Parent componentWillReceiveProps')

}

shouldComponentUpdate() {

/\*

- this method controls the state change

- return true -> change the state

- return false -> dont change the state

\*/

console.log('Parent shouldComponentUpdate')

return true

}

componentWillUpdate() {

console.log("Parent componentWillUpdate")

}

componentDidUpdate() {

console.log("Parent componentDidUpdate")

}

/\*

- Before unmounting the components react library will execute the following methods

- these methods are used to perform cleanup operations

- eg nullifying instances, empty states, empty props, cancel subscriptions, etc

\*/

componentWillUnmount() {

console.log("Parent componentWillUnmount")

}

}

\*\*\*Child.js\*\*\*

import React from "react";

export default class Child extends React.Component {

constructor() {

super()

console.log('Child constructor')

}

componentWillMount() {

console.log('Child componentWillMount called')

}

render() {

console.log('Child render called')

return (

<div>

<h1>Child Component</h1>

<h4>{this.props.key1} </h4>

</div>

)

}

componentDidMount() {

console.log('Child componentDidMount called')

}

componentWillReceiveProps() {

console.log('Child componentWillReceiveProps')

}

shouldComponentUpdate(){

console.log('Child shouldComponentUpdate')

return true

}

componentWillUpdate() {

console.log("Child componentWillUpdate")

}

componentDidUpdate() {

console.log("Child componentDidUpdate")

}

componentWillUnmount() {

console.log("Child componentWillUnmount")

}

}

==========================================================

Asynchronous calls

==========================================================

- axios library is used to make asynchronous calls

Download axios library

>yarn add axios@1.3.5 --save

Api https://bewildered-puce-wear.cyclic.app

\*\*\*GetPost.js\*\*\*

import React from 'react'

import axios from 'axios'

export default class GetPost extends React.Component {

constructor() {

super()

this.state = {

products: [],

status: ''

}

}

render() {

return (

<div className='container mt-5'>

<form onSubmit={this.insert} className='w-50 m-auto'>

<input type='number'

placeholder='p\_id'

name='p\_id'

className='form-control my-2'></input>

<input type='text'

placeholder='p\_name'

name='p\_name'

className='form-control my-2'></input>

<input type='number'

placeholder='p\_cost'

name='p\_cost'

className='form-control my-2'></input>

<input type='submit' value='Insert' className='btn btn-outline-success'></input>

</form>

<button onClick={this.getData} className='btn btn-outline-primary'>Get Data</button>

<table className='table

table-info

table-striped

table-hover

table-bordered

w-50 m-auto'>

<thead>

<th>Sr No</th>

<th>P\_id</th>

<th>P\_name</th>

<th>P\_cost</th>

</thead>

<tbody>

{this.state.products.map((e, i) => (

<tr>

<td>{i + 1} </td>

<td>{e.p\_id}</td>

<td>{e.p\_name}</td>

<td>{e.p\_cost} </td>

</tr>

))}

</tbody>

</table>

<h3>{this.state.status} </h3>

</div>

)

}

getData = () => {

this.setState({

status: 'Loading'

})

axios.get("https://bewildered-puce-wear.cyclic.app/fetch")

.then((posRes) => {

this.setState({

products: posRes.data,

status: ''

})

}, (errRes) => {

console.log(errRes)

})

}

insert = (e) => {

e.preventDefault()

let obj = {

"p\_id": parseInt(e.target.p\_id.value),

"p\_name": e.target.p\_name.value,

"p\_cost": parseInt(e.target.p\_cost.value)

}

axios.post("https://bewildered-puce-wear.cyclic.app/insert", obj)

.then((posRes) => {

console.log(posRes)

this.setState({

status: posRes.data.insert

})

}, (errRes) => {

console.log(errRes)

})

}

}

=================================================

Imports and Exports in ReactJS

=================================================

* As react developer we can export variables, objects, functions, etc
* As react developer we import any exported variables, objects, functions, etc

Note:-

1. While exporting there must be ONE member as default.

2. While exporting

export {member1, member2, member3,...}

where member1 is default.

3. If we write the default keyword, then it can be imported with any name.

Directory structure

<>

src

ios

- variables.js

- functions.js

- myComponent.js

\*\*\*variables.js\*\*\*

let url = "http://localhost:8080"

let flag = 'true'

let score = 88

let db\_config = {

host: "localhost",

user: "root",

password: "root",

database: "nodedb",

table: "products"

}

export default url

export { flag, score, db\_config }

\*\*\*functions.js\*\*\*

function fun\_one() {

return `Welcome to my function`

}

function auth(arg1, arg2) {

if (arg1 == 'admin' && arg2 == 'admin')

return true

else

return false

}

export default fun\_one

export { auth }

\*\*\*myComponent.js\*\*\*

import React from "react"

//import url, { flag, score, db\_config } from "./variables"

import \* as obj from './variables'

import fun\_one, { auth } from "./functions"

export default class MyComponent extends React.Component {

render() {

return (

<div>

{/\*<p style={{color:'rgb(255,0,0)'}}>Url :- {url}</p>

<p style={{color:'rgb(200,0,0)'}}>Flag :- {flag} </p>

<p style={{color:'rgb(180,0,0'}}>Score:- {score} </p>

<p style={{color:'rgb(160,0,0'}}>Database Configuration:- {JSON.stringify(db\_config)} </p>\*/}

<p style={{ color: 'rgb(0,255,0)' }}>Url :- {obj.default}</p>

<p style={{ color: 'rgb(0,200,0)' }}>Flag :- {obj.flag} </p>

<p style={{ color: 'rgb(0,180,0' }}>Score:- {obj.score} </p>

<p style={{ color: 'rgb(0,160,0' }}>Database Configuration:- {JSON.stringify(obj.db\_config)} </p>

<p style={{ color: 'rgb(0,0,255)' }}>{fun\_one()} </p>

<form onSubmit={this.login}>

<input type='text' placeholder='Enter Username' name='uname'></input>

<br /><br />

<input type='password' placeholder='Enter Password' name='upwd'></input>

<br /><br />

<input type='submit' value='Login'></input>

</form>

</div>

)

}

login = (e) => {

let uname = e.target.uname.value

let upwd = e.target.upwd.value

let login = auth(uname, upwd)

if (login)

alert('Login Success')

else

alert('Login Failed')

}

}

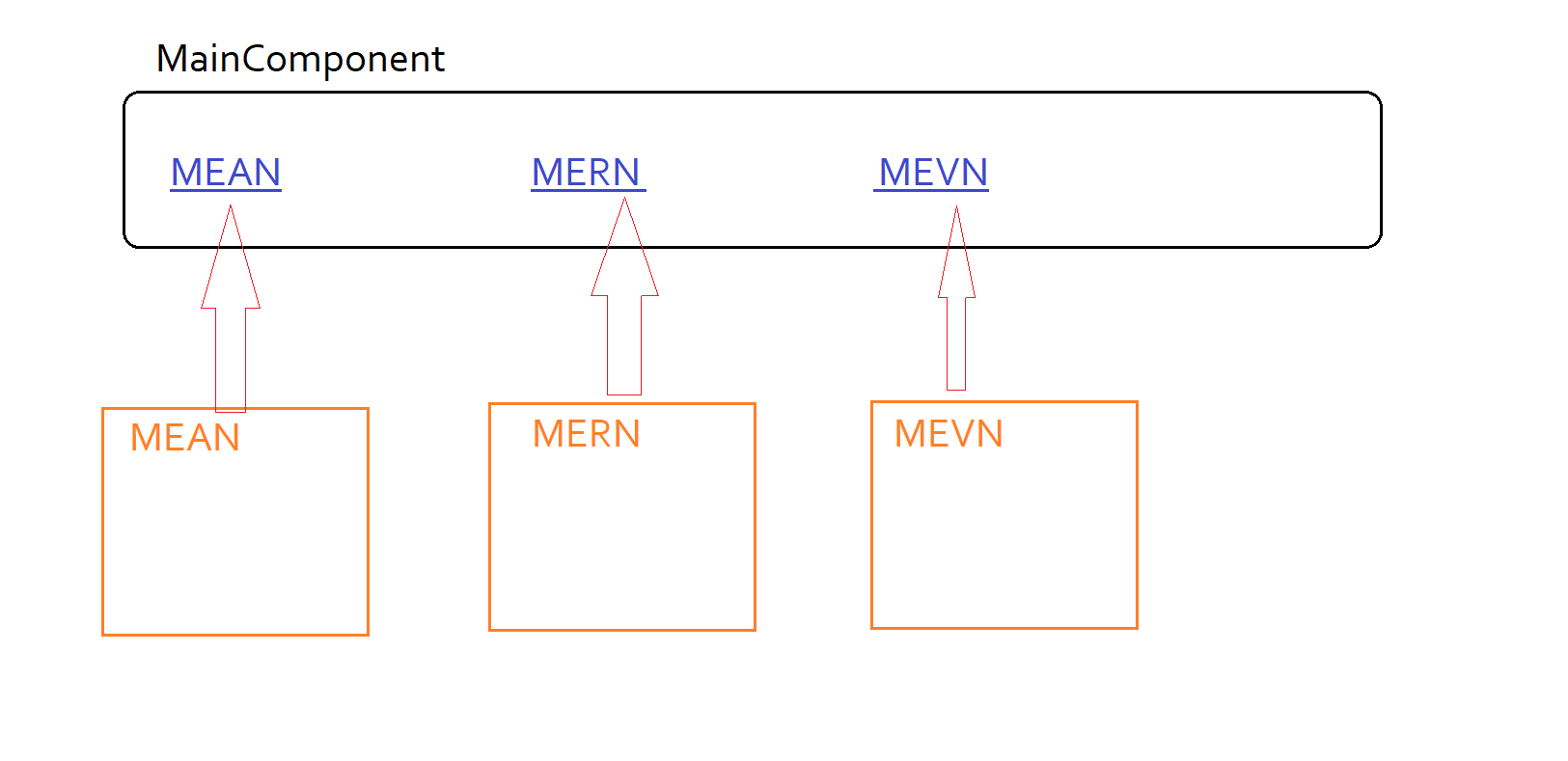
==============================================

Single Page Application

==============================================

* Loading one component to another component without refreshing the whole webpage is called a single page application.
* Navigation of components in SPA is called routing.
* 'BrowserRouter' is used to perform navigation.
* 'Route' is used to render a particular component.
* All navigations are in 'Routes'.
* 'NavLink' defines a link to particular component.
* BrowserRouter, Route, Routes and NavLink are available in 'react-router-dom'

>yarn add react-router-dom --save



Dir Stru

<>

src

SPA

- Mean.js

- Mern.js

- Mevn.js

- MainComponent.js

\*\*\*Mean.js\*\*\*

import React from 'react'

export default class MEAN extends React.Component {

render() {

return (

<div className='container mt-5'>

<h1 style={{ color: 'red' }}>Welcome to MEAN Component </h1>

</div>

)

}

}

Similarly design Mern and Mevn components

\*\*\*MainComponent.js\*\*\*

import React from 'react'

import { NavLink, BrowserRouter as Router, Routes, Route } from 'react-router-dom'

import MEAN from './Mean'

import MERN from './Mern'

import MEVN from './Mevn'

export default class MainComponent extends React.Component {

render() {

return (

<div>

<Router>

<NavLink to="/mean" style={{ marginRight: '30px' }} >MEAN </NavLink>

<NavLink to="/mern" style={{ marginRight: '30px' }} >MERN </NavLink>

<NavLink to="/mevn" style={{ marginRight: '30px' }} >MEVN </NavLink>

<Routes>

<Route path="/mean" element={<MEAN />}></Route>

<Route path='/mern' element={<MERN />}></Route>

<Route path='/mevn' element={<MEVN />}></Route>

</Routes>

</Router>

</div>

)

}

}

==============================================

Nested Routing

==============================================

\*\*\*Angular.js\*\*\*

import React from 'react'

export default class AngularComponent extends React.Component {

render() {

return (

<div>

<h2 style={{ color: 'maroon' }}>Angular </h2>

</div>

)

}

}

similarly design ReactjsComponent and VueComponent

\*\*\*Mean.js\*\*\*

import React from 'react'

import { NavLink } from 'react-router-dom'

export default class MEAN extends React.Component {

render() {

return (

<div className='container mt-5'>

<h1 style={{ color: 'red' }}>Welcome to MEAN Component </h1>

<NavLink to="angular">Angular </NavLink>

</div>

)

}

}

Similarly update Mern and Mevn components

\*\*\*MainComponent.js\*\*\*

import React from 'react'

import { NavLink, BrowserRouter as Router, Routes, Route } from 'react-router-dom'

import MEAN from './Mean'

import MERN from './Mern'

import MEVN from './Mevn'

import AngularComponent from './Angular'

import ReactjsComponent from './ReactJS'

import VeuComponent from './Veu'

export default class MainComponent extends React.Component {

render() {

return (

<div>

<Router>

<NavLink to="/mean" style={{ marginRight: '30px' }} >MEAN </NavLink>

<NavLink to="/mern" style={{ marginRight: '30px' }} >MERN </NavLink>

<NavLink to="/mevn" style={{ marginRight: '30px' }} >MEVN </NavLink>

<Routes>

<Route path="/mean" element={<MEAN />}></Route>

<Route path='/mern' element={<MERN />}></Route>

<Route path='/mevn' element={<MEVN />}></Route>

<Route path="/mean/angular" element={<AngularComponent />}></Route>

<Route path='/mern/reactjs' element={<ReactjsComponent />}></Route>

<Route path='/mevn/veu' element={<VeuComponent />}></Route>

</Routes>

</Router>

</div>

)

}

}

================================================

Lazy Loading

================================================

- Loading of Component on demand is called as Lazy Loading

- Loading time of application reduces.

- Performance of Application increases.

- Lazy Loading is used on slower internet.

import Lazy component as

const LazyComponent = lazy(() => import('./LazyComponent'))

provide navigation link as

<NavLink to="/myLazy" >Lazy</NavLink>

have a Route as

<Route path="/myLazy" element = {<Suspense fallback='Loading'><LazyComponent/> </Suspense>}></Route>

where lazy and Suspense are to be imported from 'react'

\*\*\*LazyComponent.js\*\*\*

import React from 'react'

export default class LazyComponent extends React.Component {

render() {

return (

<h1 style={{ background: 'red', color: 'white' }}>

I am from Lazy Component

</h1>

)

}

}

\*\*\*MainComponent.js\*\*\*

import React, { Suspense, lazy } from "react";

import { NavLink, BrowserRouter as Router, Routes, Route } from 'react-router-dom'

import MEAN from './Mean'

import MERN from './Mern'

import MEVN from './Mevn'

import AngularComponent from './Angular'

import ReactjsComponent from './ReactJS'

import VeuComponent from './Veu'

//---------------------------------------------//

const LazyComponent = lazy(() => import('./LazyComponent'))

//---------------------------------------------//

export default class MainComponent extends React.Component {

render() {

return (

<div>

<Router>

<NavLink to="/mean" style={{ marginRight: '30px' }} >MEAN </NavLink>

<NavLink to="/mern" style={{ marginRight: '30px' }} >MERN </NavLink>

<NavLink to="/mevn" style={{ marginRight: '30px' }} >MEVN </NavLink>

{/\*---------------------------\*/}

<NavLink to="/myLazy" >Lazy</NavLink>

{/\*---------------------------\*/}

<Routes>

<Route path="/mean" element={<MEAN />}></Route>

<Route path='/mern' element={<MERN />}></Route>

<Route path='/mevn' element={<MEVN />}></Route>

<Route path="/myLazy" element = {<Suspense fallback='Loading'><LazyComponent/> </Suspense>}></Route>

<Route path="/mean/angular" element={<AngularComponent />}></Route>

<Route path='/mern/reactjs' element={<ReactjsComponent />}></Route>

<Route path='/mevn/veu' element={<VeuComponent />}></Route>

</Routes>

</Router>

</div>

)

}

}