https://shorturl.at/3ICHe 18 Jun 2024

---

useEffect hook
useReducer hook
useCallback hook
Capstone Project Frontend (Guideline only)

==========

useEffect Hook

==========

- Consider the requirement
- provide custom title to a page instead of 'React App'
- title should be button clicked 'clicked x times'
- this is done by two lifecycle methods, componentDidMount() and componentDidUpdate().
- Consider this class component code

```
import React from 'react'
export default class Eg01 extends React.Component {
    constructor() {
        super()
        this.state = {
            count: 0
        }
    }
    componentDidMount() {
        document.title = `Clicked ${this.state.count} times`
    }
    componentDidUpdate() {
        document.title = `Clicked ${this.state.count} times`
    }
    render() {
        return (
            <div>
                <button onClick={
                    () => this.setState({ count: this.state.count + 1 })
                }>
                    Click Me
                </button>
                <br />
                Clicked {this.state.count} times
            </div>
        )
    }
}
```

- now we will do this with functional component

```
import { useEffect, useState } from "react"
export default function Eg01() {
    const [count, setCount] = useState(0)
    useEffect(() => {
        document.title = `Clicked ${count} times`
```

```
    })
    return (
        <div>
            <button
                onClick={() => setCount(count + 1)}
            >Clicked {count} times </button>
        </div>
    )
}
```

Conditional rendering

```
import React from 'react'
export default class Eg02 extends React.Component {
    constructor() {
        super()
        this.state = {
            count: 0,
            name: ''
        }
    }
    componentDidMount() {
        document.title = `Clicked ${this.state.count} times`
    }
    componentDidUpdate(prevProps, prevState) {
        if (prevState.count != this.state.count)
            console.log(`Component updated`)
        document.title = `Clicked ${this.state.count} times`
    }
    render() {
        return (
            <div>
                <button onClick={
                    () => this.setState({ count: this.state.count + 1 })
                }>
                    Click Me
                </button>
                <br />
                Clicked {this.state.count} times
                <br />
                <input type='text' onChange={e => { this.setState({ name:
e.target.value }) }}></input>
            </div>
        )
    }
}
```

```
import { useEffect, useState } from "react"
export default function Eg02() {
    const [count, setCount] = useState(0)
    const [name, setName] = useState('')
    useEffect(() => {
        console.log('useEffect Called')
        document.title = `Clicked ${count} times`
    }, [count])  //here second argument is array of values to be watched
```

```
    return (
        <div>
            <button
                onClick={() => setCount(count + 1)}
            >Clicked {count} times </button>
            <br />
            <input type='text' value={name} onChange={e =>
setName(e.target.value)}></input>
        </div>
    )
}
```

- Execute only once, no further update
- Using class component - componentDidMount()

```
import React from "react";

export default class Eg03 extends React.Component {
    constructor() {
        super()
        this.state = {
            x: 0,
            y: 0
        }
    }
    logMouse = e => {
        this.setState({
            x: e.clientX,
            y: e.clientY
        })
    }
    componentDidMount() {
        window.addEventListener('mousemove', this.logMouse)
        console.log(this.state)
    }
    render() {
        return (
            <div>
                X - {this.state.x} ... Y - {this.state.y}
            </div>
        )
    }
}

import { useState, useEffect } from 'react'
export default function Eg03() {
    const [x, setX] = useState(0)
    const [y, setY] = useState(0)
    let logMouse = e => {
        setX(e.clientX)
        setY(e.clientY)
    }
    useEffect(() => {
        console.log('useEffect Called')
        window.addEventListener('mousemove', logMouse)
    }, [])
```

```
    return (
        <div>
            X - {x} ... Y - {y}
        </div>
    )
}
```

API Calls - axios
>yarn add axios --save
use url :- https://jsonplaceholder.typicode.com/

//fetch all records

```
import { useEffect, useState } from "react";
import axios from 'axios'
export default function Eg04() {
    const [posts, setPosts] = useState([])
    useEffect(() => {
        axios.get('https://jsonplaceholder.typicode.com/posts')
            .then((posRes) => {
                setPosts(posRes.data)
            }, (errRes) => {
                console.log(errRes)
            })
    }, [])
    return (
        <div>
            <table className="table table-bordered table-warning w-75 mx-auto
table-striped table-hover text-primary">
                <thead>
                    <tr>
                        <th>Sr no</th>
                        <th>UserID</th>
                        <th>Title</th>
                    </tr>
                </thead>
                <tbody>
                    {posts.map((e, i) => (
                        <tr>
                            <td>{i + 1} </td>
                            <td>{e.userId} </td>
                            <td>{e.title}</td>
                        </tr>
                    ))}
                </tbody>
            </table>
        </div>
    )
}
```

//fetch single record based on id

```
import { useEffect, useState } from "react";
import axios from 'axios'
export default function Eg05() {
    const [posts, setPosts] = useState([])
    const [id, setId] = useState(0)
    const [nid, setNid] = useState(0)
```

```
    const fetchData = () => {
        setPosts([])
        setNid(id)
    }
    useEffect(() => {
        axios.get(`https://jsonplaceholder.typicode.com/posts/${id}`)
            .then((posRes) => {
                setPosts(posRes.data)
            }, (errRes) => {
                console.log(errRes)
            })
    }, [nid])
    return (
        <div>
            <input type='number'
                placeholder="Enter id"
                onChange={e => setId(e.target.value)}></input>
            <button onClick={fetchData}>Fetch</button>
            <h4>{JSON.stringify(posts)} </h4>
        </div>
    )
}
```

==========

useReducer

==========

```
import { useEffect, useState } from "react";
import axios from 'axios'
export default function Eg01() {
    const [data, setData] = useState([])
    const [loading, setLoading] = useState(true)
    const [error, setError] = useState('')
    useEffect(() => {
        axios.get(`https://dac930am.onrender.com/fetc`)
            .then((posRes) => {
                setLoading(false)
                setData(posRes.data)
                setError('')
            }, (errRes) => {
                setLoading(false)
                setData([])
                setError(errRes.message)
            })
    }, [])
    return (
        <div>
            <p>{loading ? 'Loading' : JSON.stringify(data)} </p>
            <p>{error ? JSON.stringify(error) : null} </p>
        </div>
    )
}
```

```
import { useEffect, useReducer } from "react"
```

```
import axios from 'axios'
const initialState = {
    data: [],
    loading: true,
    error: ''
}

const reducer = (state, actions) => {
    switch (actions.type) {
        case 'SUCCESS': {
            return {
                loading: false,
                data: actions.payload,
                error: ''
            }
        }
        case 'ERROR':
            return {
                loading: false,
                data: [],
                error: 'Error Occured ' + actions.payload
            }
    }
}

export default function Eg01(){
    const [state, dispatch] = useReducer(reducer, initialState)
    useEffect(() => {
        axios.get(`https://dac930am.onrender.com/fetch`)
            .then((posRes) => {
                dispatch({ type: 'SUCCESS', payload: posRes.data })
            }, (errRes) => {
                dispatch({ type: "ERROR", payload: errRes.message })
            })
    }, [])
    return (
        <div>
            <p>{state.loading ? 'Loading' : JSON.stringify(state.data)} </p>
            <p>{state.error ? JSON.stringify(state.error) : null} </p>
        </div>
    )
}
```

Spot differences in above two examples
- useState hook can handle simple data
- useReducer can handle complex data

==========

useCallback hook (Performance Optimization Eg01)

==========

- See the performance issue in the code

- here we passed functions as props.

Directory structure.
useCallback
- ParentCompo.js
- Title.js
- Count.js
- Button.js

Title.js

```
import React from "react"
function Title() {
    console.log('Rendering Title')
    return (
        <h2>
            useCallBack Hook
        </h2>
    )
}
//export default Title
//////////01
export default React.memo(Title)
```

Count.js

```
import React from "react"
function Count({ text, count }) {
    console.log(`Redering ${text}`)
    return <div>{text} - {count} </div>
}
//export default Count
//////////01
export default React.memo(Count)
```

Button.js

```
import React from "react"
function Button({ handleClick, children }) {
    console.log('Rendering Button - ', children)
    return (
        <button onClick={handleClick}>
            {children}
        </button>
    )
}
//export default Button
//////////01
export default React.memo(Button)
```

ParentCompo.js

```
import { useCallback, useState } from 'react'
import Title from './Title'
```

```
import Count from './Count'
import Button from './Button'
function ParentCompo() {
    const [age, setAge] = useState(25)
    const [salary, setSalary] = useState(50000)
    /*//////////02
     const incrementAge = () =>{
         setAge(age + 1)
     }

     const incrementSalary = () =>{
         setSalary(salary + 1000)
     }
     */
    //////////02
    const incrementAge = useCallback(() => {
        setAge(age + 1)
    }, [age])
    const incrementSalary = useCallback(() => {
        setSalary(salary + 1000)
    }, [salary])
    return (
        <div>
            <Title />
            <Count text="Age" count={age} />
            <Button handleClick={incrementAge}> Increment Age</Button>
            <Count text="Salary" count={salary} />
            <Button handleClick={incrementSalary}> Increment Salary</Button>
        </div>
    )
}
export default ParentCompo
```

Note:-

1. Here if we observe the console, at each button click all components are rendering.
2. To limit this rendering use React.memo
3. Now spot the issue, still there is rerendering of Button component
4. Again to optimise this we will use the 'useCallback' hook.
5. it calls function only if the dependent value i.e. second argument to useCallback changes.
6. Now the issue is solved.

Capstone Project Frontend (Guideline only)

Create a new react application
>create-react-app client
Create various components

- aboutus.js
- contactus.js
- Header.js

- MainComponent.js -> Login and Dashboard
- SignupComponent.js -> Create new user
- indexComponent.js -> Home Page
- url.js -> Backend url (APIs)

url.js

```
module.exports = "http://localhost:8080"
```

aboutus.js

```
import React from 'react'
export default class Aboutus extends React.Component{
    render(){
        return(
            <div className='container mt-5'>
                <p className='jumbotron'>Welcome to Aboutus</p>
            </div>
        )
    }
}
```

Similarly design contactus.js, MainComponent and Signup Components
Download following libraries
'react-router-dom', 'axios'
>yarn add react-router-dom axios --save
indexComponent.js

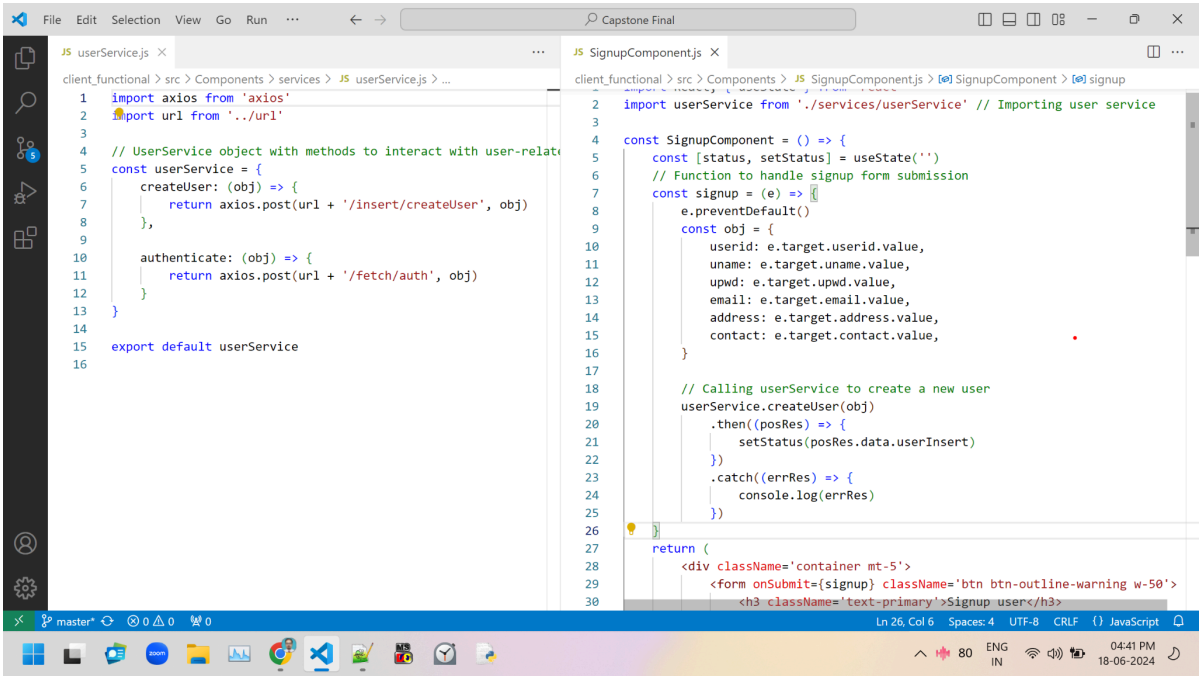
```
import React from "react";
import {NavLink, Route, BrowserRouter as Router, Routes} from
'react-router-dom'
import Aboutus from "./aboutus";
import Contactus from "./contactus";
import SignupComponent from "./SignupComponent";
import MainComponent from "./MainComponent";
export default class IndexComponent extends React.Component{
    render(){
        return(
            <div>
                <div className="nav nav-pills">
                    <Router>
                        <div className="nav-item">
                            <NavLink to = "/aboutus"
className='nav-link'>About us</NavLink>
                        </div>
                        <div className="nav-item">
                            <NavLink to = "/contactus"
className='nav-link'>Contact us</NavLink>
                        </div>
                        <div className="nav-item">
                            <NavLink to = "/signup"
className='nav-link'>Signup</NavLink>
                        </div>
                        <div className="nav-item">
```

```
                            <NavLink to = "/login"
className='nav-link'>Login</NavLink>
                        </div>
                        <br/><br/>
                        <Routes>
                            <Route path="/aboutus"
element={<Aboutus/>}></Route>
                            <Route path="/contactus" element=
{<Contactus/>}></Route>
                            <Route path="/signup"
element={<SignupComponent/>}></Route>
                            <Route path="/login"
element={<MainComponent/>}></Route>
                        </Routes>
                    </Router>
                </div>
            </div>
        )
    }
}
```

SignupComponent.js

```
import React from 'react'
import axios from 'axios'
import url from './url'
export default class SignupComponent extends React.Component {
    constructor() {
        super()
        this.state = {
            status:''
        }
    }
    render() {
        return (
            <div className='container mt-5'>
                <form onSubmit={this.signup} className='btn
btn-outline-warning w-50'>
                    <h3 className='text-primary'>Signup user </h3>
                    <div className='form-group my-2 btn btn-outline-dark p-3
w-100'>
                        <label>User id</label>
                        <input type='text' placeholder='Enter Userid'
className='form-control' name='userid'></input>
                    </div>
                    <div className='form-group my-2 btn btn-outline-dark p-3
w-100'>
                        <label>User Name</label>
                        <input type='text' placeholder='Enter User Name'
className='form-control' name='uname'></input>
                    </div>
                    <div className='form-group my-2 btn btn-outline-dark p-3
w-100'>
                        <label>Password</label>
```

```
                        <input type='password' placeholder='Enter Password'
className='form-control' name='upwd'></input>
                    </div>
                    <div className='form-group my-2 btn btn-outline-dark p-3
w-100'>
                        <label>User email</label>
                        <input type='email' placeholder='Enter User email'
className='form-control' name='email'></input>
                    </div>
                    <div className='form-group my-2 btn btn-outline-dark p-3
w-100'>
                        <label>User Address</label>
                        <input type='text' placeholder='Enter User Address'
className='form-control' name='address'></input>
                    </div>
                    <div className='form-group my-2 btn btn-outline-dark p-3
w-100'>
                        <label>Contact</label>
                        <input type='text' placeholder='Enter Contact'
className='form-control' name='contact'></input>
                    </div>
                    <div className='form-group my-2 w-25 mx-auto' align =
'center'>
                        <input type='submit' className='btn
btn-outline-success' value='Signup'></input>
                        <h3>{this.state.status}</h3>
                    </div>
                </form>
            </div>
        )
    }
    signup = (e) => {
        e.preventDefault()
        let obj = {
            "userid" : e.target.userid.value,
            "uname" : e.target.uname.value,
            "upwd" : e.target.upwd.value,
            "email" : e.target.email.value,
            "address" : e.target.address.value,
            "contact" : e.target.contact.value
        }
        axios.post(url+"/insert/createUser",obj)
            .then((posRes)=>{
                console.log(posRes.data)
                this.setState({
                    status : posRes.data.userInsert
                })
            },(errRes)=>{
                console.log(errRes)
            })
    }
}
```


File Edit Selection View Go Run
Capstone Final
userService.js
client_functional > src > Components > services > userService.js > ...
import axios from 'axios'
import url from '../url'
// UserService object with methods to interact with user-relate
const userService = {
createUser: (obj) => {
return axios.post(url + '/insert/createUser', obj)
},
authenticate: (obj) => {
return axios.post(url + '/fetch/auth', obj)
}
}
export default userService
SignupComponent.js
client_functional > src > Components > SignupComponent.js > SignupComponent > signup
import userService from './services/userService' // Importing user service
const SignupComponent = () => {
const [status, setStatus] = useState('')
// Function to handle signup form submission
const signup = (e) => {
e.preventDefault()
const obj = {
userid: e.target.userid.value,
uname: e.target.uname.value,
upwd: e.target.upwd.value,
email: e.target.email.value,
address: e.target.address.value,
contact: e.target.contact.value,
}
// Calling userService to create a new user
userService.createUser(obj)
.then((posRes) => {
setStatus(posRes.data.userInsert)
})
.catch((errRes) => {
console.log(errRes)
})
}
return (

<form onSubmit={signup} className='btn btn-outline-warning w-50'>
<h3 className='text-primary'>Signup user</h3>
master*
Ln 26, Col 6
Spaces: 4
UTF-8
CRLF
JavaScript
ENG
IN
04:41 PM
18-06-2024