React Workshop tutorial-React juni 2022

Server start in localhost:3030 https://github.com/softuni-practice-server/softuni-practice-server

comand: node server

1.Create react app

1.1 npx create-react-app my-app 1.2 cd D:/Java-script/my-app 1.3 npm start

2.Renname all class to className in HTML templates

3. Create folder components

4. Strart create components

5.Start create Router

5.1 npm install react-router-dom

5.2 use browser router in index.js

<BrowserRouter>

    <App />

</BrowserRouter>

5.3 use routes in App.js

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

import { Header } from "./components/Header/Header";

import { HomePage } from "./components/HomePage/HomePage";

function App() {

  return (

    <div id="box">

      <Header />

      <main id="main-content">

        <Routes>

          <Route path='/' element={<HomePage />}/>

        </Routes>

      </main>

    </div>

  );

}

export default App;

6.Replase <a>Hello</a> tag with <Link>Hello</Link> tag

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

import { useContext } from 'react'

import { AuthContext } from '../../contexts/AuthContext'

export const Header = () => {

    const { auth } = useContext(AuthContext)

    return (

        <header>

            <h1>

                <Link to={'/'}>GamesPlay</Link>

            </h1>

            <nav>

7.Create context in gameContext.js file

import { createContext, useEffect, useState } from 'react'

import \* as services  from '../services/services'

export const GameContext = createContext();

export const GameProvider = ({

    children

}) => {

    const [games, setGames ]= useState([])

    useEffect(() =>{

      services.getAll()

      .then(result => {

        setGames(result)

      })

    },[])

    return (

      <GameContext.Provider value={{games, }}>

        {children}

      </GameContext.Provider>

    )

}

8.Set returned GameContext.Provider in App.js

function App() {

  return (

    <div id="box">

      <Header />

      <GameProvider>

        <main id="main-content">

          <Routes>

            <Route path='/' element={<HomePage />} />

            <Route path='/login' element={<Login />} />

            <Route path='/register' element={<Register />} />

            <Route path='/details/:id' element={<DetailsPage />} />

            <Route path='/edit/:id' element={<EditPage />} />

            <Route path='/create' element={<CreatePage />} />

            <Route path='/catalog' element={<Catalog />} />

          </Routes>

        </main>

      </GameProvider>

    </div>

  );

}

export default App;

9.Use context in another files

import { useContext } from "react"

import { GameCard } from "./GameCard"

import { GameContext } from "../../contexts/gameContext"

export const HomePage = () => {

    const {games} = useContext(GameContext)

    return (

        <section id="welcome-world">

            <div className="welcome-message">

                <h2>ALL new games are </h2>

                <h3>Only in GamesPlay</h3>

            </div>

            <img src="./images/four\_slider\_img01.png" alt="hero" />

10.Fetch all games with fetch api in services.js

export const getAll = async() => {

    const res = await fetch(url)

    const result = await res.json()

    return result

}

11.Use games in contexts.js

import \* as services  from '../services/services'

const [games, setGames ]= useState([])

    useEffect(() =>{

      services.getAll()

      .then(result => {

        setGames(result)

      })

    },[])

    return (

      <GameContext.Provider value={{games, }}>

        {children}

      </GameContext.Provider>

    )

12.Use games in Catalog.js and give the game in <AllGames /> via props

import { useContext } from 'react'

import { GameContext } from '../../contexts/gameContext'

import { AllGames } from './AllGames'

export const Catalog = () => {

    const {games} = useContext(GameContext)

    return (

        <section id="catalog-page">

        <h1>All Games</h1>

       { games.length > 0

       ? games.map(game => <AllGames key={game.\_id} game={game}/> )

       : <h3 className="no-articles">No articles yet</h3>

       }

    </section>

    )

}

13.Recive game via props and set id in details url

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

export const AllGames = ({

    game

}) => {

    return (

        <div className="allGames">

            <div className="allGames-info">

                <img src={game.imageUrl} alt="img"/>

                <h6>{game.category}</h6>

                <h2>{game.title}</h2>

                <Link to={`/details/${game.\_id}`} className="details-button">Details</Link>

            </div>

        </div>

    )

}

14.Use hook useParams, useNavigate in DetailsPage.js

import { useEffect, useState } from 'react'

import { Link, useParams, useNavigate } from 'react-router-dom'

import \* as services from '../../services/services'

export const DetailsPage = () => {

    const {id} = useParams() useParams() return id like string **!!!!!**

    const [game, setGame] = useState({})

    const navigate = useNavigate()

    useEffect(() => {

        services.getOne(id)

        .then(result => {

            setGame(result)

        })

    }, [id])

    const onDelete = () => {

        console.log('ok')

        services.deleteOne(id)

        navigate('/catalog')

    }

15.Use fetch api async and non async in services.js

const url = 'http://localhost:3030/data/games'

// const url = 'http://localhost:8000/games'

// export const getAll = () => {

//     return fetch(url)

//         .then(res => res.json())

// }

// export const getOne = (id) => {

//     return fetch(`${url}/${id}`)

//         .then(res => res.json())

// }

// export const deleteOne = (id) => {

//     return fetch(`${url}/${id}`, {

//         method: 'DELETE',

//     })

//         .then(res => res.json())

// }

export const getAll = async() => {

    const res = await fetch(url)

    const result = await res.json()

    return result

}

export const getOne = async (id) => {

    const res = await fetch(`${url}/${id}`)

    const result = await res.json()

    return result

}

export const deleteOne = async (id) => {

    const res = await fetch(`${url}/${id}`, {

        method: 'DELETE',

    })

    const result = await res.json()

    return result

}

16.Make delete functionality in DetailsPage

  const onDelete = () => {

        services.deleteOne(id)

        delGame(id)

        navigate('/catalog')

    }

  <div className="buttons">

                <Link to={`/edit/${id}`} className="button">Edit</Link>

                <button className="button" onClick={onDelete}>Delete</button>

            </div>

17. Create delGame function in gameContext.js

    const [games, setGames ]= useState([])

    useEffect(() =>{

      services.getAll()

      .then(result => {

        setGames(result)

      })

    },[])

    const delGame = (id) => {

      setGames(state => (state.filter(obj => obj.id !== parseInt(id))))

    }

18. start with CreatePage

18.1 set onSubmit in form

 <section id="create-page" className="auth">

            <form id="create" onSubmit={onSubmit}>

                <div className="container">

18.2 Create onSubmit function and extract data from form and navigate to CatalogPage

 const onSubmit = (ev) => {

        ev.preventDefault();

        const gameData = Object.fromEntries(new FormData(ev.target))

        services.createOne(gameData)

            .then(res => addGame(res))

  navigate('/catalog')

    }

18.3 create addGame function in gameContext

 const addGame = (game) => {

      setGames(games => [...games, game])

    }

18.4 create createOn function in services

export const createOne = async (data) => {

    const res = await fetch(url, {

        method: 'POST',

        headers: {

            'Content-Type': 'application/json'

        },

        body: JSON.stringify(data)

    })

    const result = await res.json()

    return result

}

19.Create EditPage fuctionality

19.1 Use useParams and useState to extract id and set current state for game

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

import { useEffect, useState } from 'react'

import \* as services from '../../services/services'

export const EditPage = () => {

    const {id} = useParams()

    const [game, setGame] = useState({

        title: '',

        category: '',

        maxLevel: '',

        imageUrl: '',

        summary: ''

    })

19.2 take the game from server with useEfect and getOne

   useEffect(() => {

       services.getOne(id)

       .then(res => setGame({

        title: res.title,

        category: res.category,

        maxLevel: res.maxLevel,

        imageUrl: res.imageUrl,

        summary: res.summary

       }))

    },[id])

19.3 create onChange functtion and set in all inputs in form

const onChange = (ev) => {

        setGame(state => ({

            ...state,

           [ev.target.name]: ev.target.value

        }))

    }

    return (

        <section id="edit-page" className="auth">

        <form id="edit">

            <div className="container">

                <h1>Edit Game</h1>

                <label htmlFor="leg-title">Legendary title:</label>

                <input

                type="text"

                id="title"

                name="title"

                value={game.title}

                onChange={onChange}

                />

                <label htmlFor="category">Category:</label>

                <input

                type="text"

                id="category"

                name="category"

                value={game.category}

                onChange={onChange}

                />

                <label htmlFor="levels">MaxLevel:</label>

19.4 create onSubmit function and set it in form and use useNavigate to redirect to Details

 const onSubmit = (ev) => {

        ev.preventDefault()

        services.editOne(id, game)

        .then(res => {

            editGame(id, res)

        })

        navigate(`/details/${id}`)

    }

    return (

        <section id="edit-page" className="auth">

        <form id="edit" onSubmit={onSubmit}>

            <div className="container">

19.5 create editGame function in GameContext

 const editGame = (id, game) => {

      setGames(state => (state.map(obj => obj.id !== parseInt(id) ? obj : game)))

    }

20. create the last games functionality in home page

 <div id="home-page">

                <h1>Latest Games</h1>

                { lastGames.length > 0

                ? lastGames.map((game, idx) =>  <GameCard  key={game.id} game={game}/>)

                : <p className="no-articles">No games yet</p>

                 }

            </div>

21. create register page functionality

export const Register = () => {

    const navigate = useNavigate()

    const onSubmit = (ev) => {

        ev.preventDefault();

        const registerData = Object.fromEntries(new FormData(ev.target))

        userServices.userRegister(registerData)

        .then(res => console.log(res))

        ev.target.reset()

        navigate('/')

}

21.1 create function userRegister in service.js in userServices.js

export const userRegister = async (data) => {

    const res = await fetch(registerUrl, {

        method: 'POST',

        headers: {

            'Content-Type': 'application/json'

        },

        body: JSON.stringify(data)

    })

    const result = await res.json()

    return result

}

22. create login page functionality

export const Login = () => {

    const navigate = useNavigate()

    const {loginUser} = useContext(AuthContext)

    const onSubmit = (ev) => {

        ev.preventDefault();

        const loginData = Object.fromEntries(new FormData(ev.target))

        userServices.logUser(loginData)

        .then(res => loginUser(res))

        ev.target.reset()

        navigate('/')

    }

22.1 create function userLogin in service.js in userServices.js

export const logUser = async (data) => {

    const res = await fetch(loginUrl, {

        method: 'POST',

        headers: {

            'Content-Type': 'application/json'

        },

        body: JSON.stringify(data)

    })

    const result = await res.json()

    return result

}

23. create custom hook useLocalStorage

import { useState } from "react";

export const useLocalStorage = (key, defaultValue) => {

  const [value, setValue] = useState(() => {

    const storedData = localStorage.getItem(key)

    return storedData ? JSON.parse(storedData) : defaultValue

  })

  const setLocalStorageValue = (newValue) => {

      localStorage.setItem(key, JSON.stringify(newValue))

      setValue(newValue)

  }

  return [

    value, setLocalStorageValue

  ]

}

24. create authContext

import { createContext } from "react";

import { useLocalStorage } from "../hooks/useLocalStorage"

export const AuthContext = createContext();

export const AuthProvider = ({

    children

}) => {

    const [auth, setAuth] = useLocalStorage('auth', {})

    const userLogin = (authData) => {

        setAuth(authData)

    }

    const userLogout = () => {

        setAuth({})

    }

    return (

    <AuthContext.Provider value={{auth, userLogin, userLogout}}>

        {children}

    </AuthContext.Provider>

    )

}

25. create Logout page functionality

import { useContext, useEffect } from "react";

import { AuthContext } from "../../contexts/authContext";

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

import \* as userServices from '../../services/userServices'

export const Logout = () => {

    const {auth, userLogout} = useContext(AuthContext)

    const navigate = useNavigate()

    const token = auth.token

    useEffect(() => {

        userServices.logoutUser(token)

        .then(status => {

            if (status === 204) {

                userLogout()

            }

        })

        navigate('/')

    })

}

25.1 create function logoutUser in userServices.js

export const logoutUser = async (token) => {

    const res = await fetch(logoutURL, {

        method: 'POST',

        headers: {

            'Authorization': `Token ${token}`

        },

    })

    return res.status

}

25.2 create function userLogout in authContext.js

const userLogout = () => {

        setAuth({})

    }