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()

    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

}