useState hook

import { useState } from "react"

const Timer = (props) => {

    const [seconds, setSeconds] = useState(props.start)

    setTimeout(() => {

        // setSeconds(seconds + 1) два начина за използване

        setSeconds(state => state + 1 )

    }, 1000);

    return (

        <div>

            <h4>State Demo</h4>

            <p>Timer: {seconds}s</p>

        </div>

    )

}

export default Timer

event in React and inline styling

import { useState } from "react"

const Counter = () => {

    const [count, setCount] = useState(0)

    const addHandler = () => {

        setCount(count + 1)

    }

    const subHandler = () => {

        setCount(count - 1)

        if (count <= 0) {

            setCount(0)

        }

    }

    return ( double curly brackets

        <div>

            <p>Count: {count}</p>

            <button style={{marginRight: "10px"}} onClick={subHandler}>-</button>

            <button onClick={addHandler}>+</button>

        </div>

    )

}

Destructuring object to props

import Movie from "./Movie"

тук деструктурираме обекта с къдрави скоби

function MovieList({movies}) {

    const firstMovie = movies[0]

    // '''THIS IS firstMovie ON OBJECT'''

    // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    // const firstMovie = {

    //     "title": "The Cotton Club",

    //     "year": "1984",

    //     "director": "Francis Ford Coppola",

    //     "plot": "The Cotton Club was a famous night club in Harlem. The story follows the people that visited the club, those that ran it, and is peppered with the Jazz music that made it so famous.",

    //     "posterUrl": "https://images-na.ssl-images-amazon.com/images/M/MV5BMTU5ODAyNzA4OV5BMl5BanBnXkFtZTcwNzYwNTIzNA@@.\_V1\_SX300.jpg"

    // }

    // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    return (

        <div>

            <h2>This coming from MovieList2</h2>

            <Movie {...firstMovie}/>

        </div>

    )

Second exempel destructuring props

import MovieList from './components/MovieList';

import './App.css';

import { movies } from './movies';

import Numbers from './components/Numbers';

function App() {

  const numbers = [1, 2, 3, 4]

  return (

    <div className="App">

      <h1>Movie collection</h1>

      <Numbers numbers={numbers}/>

      <MovieList movies={movies.slice(0, 10)}/>

    </div>

  );

}

export default App;

подаваме props

// function Numbers(props) {

//     return (

//         <ul>

//             {props.numbers.map((num) => <li>{num}</li>)}

//         </ul>

//     )

// }

Деструктурираме за да не използваме props.numbers

function Numbers({numbers}) {

    return (

        <ul>

            {numbers.map((num) => <li>{num}</li>)}

        </ul>

    )

}

export default Numbers

wrapper functuon in arrow function

function Movie({

    id,

    title,

    year,

    director,

    plot,

    posterUrl,

    onMovieDelete получаваме фукцията onMovieDelete като props

}) {

    return (

        <li>

            <h3>{title}, {year}</h3>

            <main>

                <p>{plot}</p>

                <img src={posterUrl} alt={title} />

            </main>

            <footer>

                <h4>director: {director}</h4>

                <button onClick={() => onMovieDelete(id)}>Delete</button>

            </footer>

        </li> рапваме я в аро фунция

    )

}

export default Movie

Fragment from react

import { Fragment } from "react"

export const Test = (user) => {

    return (

      <> <Fragment> this parent element not showing in DOM

            <td>

              <img src={user.imageUrl} alt="Peter's profile" className="image" />

            </td>

            <td>{user.firstName}</td>

            <td>{user.lastName}</td>

            <td>{user.email}</td>

            <td>{user.phoneNumber}</td>

            <td>{user.createdAt.split('T')[0]}</td>

      </> </Fragment>

    ) this syntax is the same like <Fragment>

}

Use services

<!-- This is name of the file './services/userServices' -->

export const getAll = async() => {

    const res = await fetch('http://localhost:3005/api/users')

    const result = await res.json()

    return result.users

}

import { useEffect, useState } from 'react';

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

import { Footer } from "./components/common/Footer";

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

import { Search } from "./components/search/Search";

import { UserList } from "./components/userList/UserList";

function App() {

  const [users, setUsers] = useState([])

  useEffect(() => {

     userService.getAll()

     .then(result => setUsers(result)) импортваме функцията getAll()

и получаваме резултата в .then()

  }, [])

Take data form Form

  const onUserCreate = (e) => {

        e.preventDefault()

        const formData = new FormData(e.target) връща масив от двойни масиви[

        const { [firstName, Konstantin],

            firstName, [lastName, Kostov],..

            lastName, ] name, value на инпут полето

            email,

            phoneNumber, връща обект { firstName: Konstantin,

lastName: Kostov,.. }

        } = Object.fromEntries(formData)

        console.log(firstName, lastName, email, phoneNumber)

    }

Destructuring json object from server

// this object coming from server

const objectFromServer = {

    "firstName": "John",

    "lastName": "Dow",

    "email": "ivan@abv.bg",

    "imageUrl": "http//google.com",

    "phoneNumber": "0889765432",

    "country": "Bulgaria",

    "city": "Varna",

    "street": "Solo",

    "streetNumber": "15"

}

// destructuring in first five variables the rest data going in variable address some contains object

const {

    firstName,

    lastName,

    email,

    imageUrl,

    phoneNumber,

    ...address } = objectFromServer

// make new object from all variables

const newObject = {

    firstName,

    lastName,

    email,

    imageUrl,

    phoneNumber,

    address

}

// this is the new object

const newFormat = {

    firstName: 'Johnny',

    lastName: 'Dow',

    email: 'ivan@abv.bg',

    imageUrl: 'http//google.com',

    phoneNumber: '0889765432',

    address: {

        country: 'Bulgaria',

        city: 'Varna',

        street: 'Solo',

        streetNumber: '15'

    }

}

Validate form library

<https://formik.org/>

load spiner

<li>{!starShips.length && '...load'}</li>

Инсталиране на router библиотека

nmp I react-router-dom

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*инсталиране на uniqid библиотека

PS D:\Java Script\React-JSX\Routing-Workshop\games-play> npm i uniqid

Същата работа върши import {useId} from ‘react’

import React from 'react';

import ReactDOM from 'react-dom/client';

import {BrowserRouter} from 'react-router-dom' импортваме BrowserRouter

import App from './App';

import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

 <BrowserRouter> поставяме App компонента вътре в BrowserRouter

    <App />

 </BrowserRouter>

);

reportWebVitals();

подаване на функция от горе на долу през компонентите

  const taskRemoveHandler = (taskId) => {

    console.log(taskId)

    setTasks(state => state.filter(obj => obj.\_id !== taskId))

  }

  return (

    <div className='site-wrapper'>

      <header>

        <h1>TODO App</h1>

      </header>

      <main>

        <TaskList tasks={tasks} taskRemoveHandler={taskRemoveHandler} />

        <CreateTask taskCreateHandler={taskCreateHandler}/>

      </main>

    </div>

  );

import TaskItem from "./TaskItem"

const TaskList = ({ tasks, taskRemoveHandler }) => {

    return (

        <ul>

            {tasks.map(task =>

                <TaskItem key={task.\_id} task={task}

                    taskRemoveHandler={taskRemoveHandler}

                />

            )}

        </ul>

    )

}

export default TaskList

const TaskItem = ({ task, taskRemoveHandler }) => {

    return (

        <div className="list-item" id={task.\_id}>

            <li>

                {task.title}

            </li>

            <button onClick={() => taskRemoveHandler(task.\_id)}>x</button>

        </div>

    )

}

export default TaskItem

use higher order function in react

  useEffect(() => {

        console.log('Mount')

        return () => {

            console.log('Unmount');

        }

    },[])

Context

First step create context

import {createContext} from 'react'

export const TaskContext = createContext()

second step import context

import "./App.css";

import { TaskContext } from './contexts/TaskContext';

import { useFetch } from './hooks/useFetch';

third step set context,give provider and value

  <TaskContext.Provider value={'Pesho'}>

            <div className='site-wrapper'>

                <header>

                    <h1>TODO App</h1>

                </header>

                <main>

                    {isLoading

                        ? <TaskList tasks={tasks}

taskRemoveHandler={taskRemoveHandler} />

                        :

                        <h3>Loading...</h3>

                    }

                    <CreateTask taskCreateHandler={taskCreateHandler} />

                </main>

            </div>

    </TaskContext.Provider>

Fourth step use context with hook useContext

import { useEffect , useContext} from "react"

import { TaskContext } from "../contexts/TaskContext"

const TaskItem = ({

    task,

    taskRemoveHandler

}) => {

    const value = useContext(TaskContext)

    console.log(value)

    return (

        <div className="list-item" id={task.\_id}>

            <li>

                {task.title}

            </li>

            <button onClick={() => taskRemoveHandler(task.\_id)}>x</button>

        </div>

    )

}