Bitya V2 react + redux

Contents

[Create a new project 1](#_Toc89077361)

[Extensions: 1](#_Toc89077362)

[Better Comments 1](#_Toc89077363)

[Material Icon Theme 1](#_Toc89077364)

[Toggle Zen mode 1](#_Toc89077365)

[Bracket Pair Colorizer 1](#_Toc89077366)

[snippets : 1](#_Toc89077367)

[UseState -first Demo (counter) 2](#_Toc89077368)

[**Fibonacci** demo of useState 2](#_Toc89077369)

[Basic Tic-Tac-Toe (useState ,UseEffect) 3](#_Toc89077370)

[map 6](#_Toc89077371)

[spread operator 7](#_Toc89077372)

[Props 7](#_Toc89077373)

[Forms 10](#_Toc89077374)

[Bootstrap 11](#_Toc89077375)

[Bootstrap 12](#_Toc89077376)

# Create a new project

npx create-react-app my-app (or npx create-react-app my-app --template typescript)

# Extensions:

## Better Comments

## Material Icon Theme

## Toggle Zen mode

## Bracket Pair Colorizer

# snippets :

imr - import react from..

rafce - react function component

# UseState -first Demo (counter)

import "./App.css";

import React, { useState } from "react";

function App() {

  //counter - state

  //setcounter - logic (update the state -react is aware)

  // in reaction the react update the GUI when we update the state with setcounter

  const [counter, setcounter] = useState(8);

  // const rnd = () => Math.random() \* 76;

  return (

    <div>

      <button onClick={() => setcounter(counter + 1)}>{counter}</button>

    </div>

  );

}

export default App;

# [**Fibonacci**](https://stackoverflow.com/questions/43949677/fibonacci-counter-in-react-js) demo of useState

import React, { useState } from "react";

const App = () => {

  const [fib, setFib] = useState([0, 1]);

  // [1,2]=> [2,3]

  return (

    <div>

      <button onClick={() => setFib([fib[1], fib[0]+ fib[1]])}>

        {fib[0] + fib[1]}

      </button>

    </div>

  );

};

export default App;

# Basic Tic-Tac-Toe (useState ,UseEffect)

import React, { useState, useEffect } from "react";

const App = () => {

  return (

    <div>

      <MyBoard />

    </div>

  );

};

export default App;

// end of App Component

export const MyBoard = () => {

  // to create init button array

  const butAr = [1, 2, 3, 4, 5, 6, 7, 8, 9];

  //board status

  const [boardStatus, setStatus] = useState([0, 0, 0, 0, 0, 0, 0, 0, 0]);

  //which player turn

  const [turn, setturn] = useState(false);

  useEffect(() => {

    if (turn) {

      localStorage.setItem("boardStatus", boardStatus);

      console.log(Math.random() \* 20);

      console.log(turn);

    }

  });

  const handleClick = (t, btnStatus) => {

    //check if button  clicked

    if (btnStatus != 0) return;

    setturn(!turn);

    //update board status

    boardStatus[t.id] = turn ? "O" : "X";

    setStatus(boardStatus);

    // checkWin()

    console.log(turn);

  };

  return (

    <>

      MyBoard Turn {Math.random() \* 20}

      {!turn ? (

        <span style={{ color: "red" }}> X</span>

      ) : (

        <span style={{ color: "blue" }}> O</span>

      )}

      <div style={{ width: 80 }}>

        {butAr.map((x, index) => (

          <Square

            key={x} id={index} handleClick={handleClick} display={boardStatus[index]}

          ></Square>

        ))}

      </div>

    </>

  );

};

export const Square = (props) => {

  return (

    <button

      id={props.id} onClick={(e) => props.handleClick(e.target, props.display)}    >

      {props.display}

    </button>

  );

};

second solution

import React, { useState } from "react";

const App = () => {

  const [fib, setFib] = useState([0, 1]);

  const calc = (n1, n2) => {

    setFib([n2, n1 + n2]);

  };

  // [1,2]=> [2,3]

  return (

    <div>

      <button onClick={() => calc(fib[0], fib[1])}>{fib[0] + fib[1]}</button>

    </div>

  );

};

export default App;

map – method to iterate the array…

import React from "react";

const App = () => {

  const myArray = ["apple", "banana", "orange"];

  return (

    <div>

          <h1>my Array </h1>

          {myArray.map((x, ind) => (

            <li key={ind}>{x}</li>

          ))}

    </div>

  );

};

export default App;

Map with simple condition

import React from "react";

const App = () => {

  const myArray = ["apple", "banana", "orange"];

  return (

    <div>

      {myArray.length > 0 && (

        <div>

          <h1>my Array </h1>

          {myArray.map((x, ind) => (

            <li key={ind}>{x}</li>

          ))}

        </div>

      )}

    </div>

  );

};

export default App;

# spread operator

 const ar = [2, 4, 5, 7];

  const test = () => {

    myFunc(ar);

    const newAr = [3, ...ar];

    myFunc(newAr);

    myFunc(2, 4, 5, 6, 7);

    myFunc(...ar);

  };

  const myFunc = (...x) => {

    console.log(x);

  };

Props - (same as input in angular)

Allow us to pass any value to child component as attribute

In the app.js (parent)

<MyDiv colorrrr={clr} index={ind} key={ind}></MyDiv>

Child

import React from "react";

const MyDiv = (props) => {

  return (

    <div style={{ padding: 100, backgroundColor: props.colorrrr }}>

      {console.log(props)}

      myDiv {props.index}

    </div>

  );

};

export default MyDiv;

example 2

parent

import React from "react";

import MyRow from "./MyRow";

const App = () => {

  const colorsAr = ["red", "green", "blue"];

  const team = ["maya", "yarden", "avital", "gal"];

  return (

    <div>

      <h1 style={{ color: "red" }}>My Team</h1>

      <table border="1">

        {team.map((teamMember, ind) => (

          <MyRow

            teamMember={teamMember}

            key={ind}

            clr={colorsAr[ind % colorsAr.length]}

            ind={ind}

          ></MyRow>

        ))}

      </table>

    </div>

  );

};

export default App;

child

import React from "react";

const MyRow = (props) => {

  return (

    <tr>

      <td style={{ color: props.clr }}>{props.ind}</td>

      <td style={{ backgroundColor: props.clr }}>{props.teamMember}</td>

      <td style={{ color: props.clr }}>{props.teamMember}</td>

      <td>

        {props.teamMember.length > 4 ? (

          <input />

        ) : (

          <button>Del: {props.teamMember}</button>

        )}

      </td>

    </tr>

  );

};

export default MyRow;

Example 3

import React from "react";

import MyButton from "./MyButton";

const App = () => {

  const colorsAr = ["red", "green", "blue"];

  const team = ["maya", "yarden", "avital", "gal"];

  const teamMemberSelect = (tmMember) => {

    console.log(tmMember);

  };

  return (

    <div>

      <table border="1">

        {team.map((teamMember, ind) => (

          <MyButton

            teamMember={teamMember}

            selectMember={teamMemberSelect}

            clr={colorsAr[ind % colorsAr.length]}

          ></MyButton>

        ))}

      </table>

    </div>

  );

};

export default App;

child

import React from "react";

const MyButton = (props) => {

  return (

    <button

      onClick={() => props.selectMember(props.teamMember)}

      style={{ color: props.clr }}

    >

      {props.teamMember}

    </button>

  );

};

export default MyButton;

**Practice (HW)**

Tic-tac-toe (Map, Props, JS vanilla)

App – win counting (X-4, O-3)

Board – check win, who’s turn, Msg

Square – get event from user, the GUI

\*Bootstrap, toast

# Forms

Every field that we want to manage (state, send to server) need 2 properties

onChange={(e) => setName(e.target.value)} – update the state from input

value={name}  - update input from state

(e) - param

(e) – Event (all the properties of the current events)

e.target – the tag that fire the event

e.target.value – the value of the current input

import React, { useState } from "react";

const App = () => {

  const [name, setName] = useState("");

  return (

    <form>

      <label>

        Enter your name:

        <input  value={name}  onChange={(e) => setName(e.target.value)}

        />

      </label>

    </form>

  );

};

export default App;

Example 2

import React, { useState } from "react";

const App = () => {

  const [name, setName] = useState("");

  const [LastName, setLastName] = useState("");

  const handleSubmit = (e) => {

    e.preventDefault();

    console.log(`The name you entered was: ${name} Last name ${LastName}`);

  };

  return (

    <form onSubmit={handleSubmit}>

      <label>

        Enter your name:

        <input value={name} onChange={(e) => setName(e.target.value)} />

      </label>

      <br />

      <label>

        Enter your Last name:

        <input value={LastName} onChange={(e) => setLastName(e.target.value)} />

      </label>

      <br />

      <input type="submit" />

    </form>

  );

};

export default App;

HW – get from user name + favorite color (select)

On submit – console “your favorite….”

# Bootstrap

https://react-bootstrap.github.io/

# Bootstrap

Adding a button

npm install react-bootstrap bootstrap@5.1.3

import Button from "react-bootstrap/Button";

import "bootstrap/dist/css/bootstrap.min.css";

# Routing -simple

import React from "react";

import { BrowserRouter as Router, Switch, Route, Link } from "react-router-dom";

const App = () => {

  return (

    <Router>

      <div>

{/\*create a menu for the user -display links. \*/}

        <nav>

          <ul>

            <li>

              <Link to="/">Home</Link>

            </li>

            <li>

              <Link to="/about">About</Link>

            </li>

            <li>

              <Link to="/users">Users</Link>

            </li>

          </ul>

        </nav>

        {/\* A <Switch> looks through its children <Route>s and

        renders the first one that matches the current URL. \*/}

        <Switch>

          <Route path="/about">

            <About />

          </Route>

          <Route path="/users">

            <Users />

          </Route>

          <Route path="/">

            <Home />

          </Route>

        </Switch>

      </div>

    </Router>

  );

};

const Home = () => <h2>Home</h2>;

function About() {

  return <h2>About</h2>;

}

function Users() {

  return <h2>Users</h2>;

}

export default App;

# Nested routing (V.5)

import React from "react";

import {

  BrowserRouter as Router,

  Switch,

  Route,

  Link,

  useParams,

  useRouteMatch,

} from "react-router-dom";

// Since routes are regular React components, they

// may be rendered anywhere in the app, including in

// child elements.

//

// This helps when it's time to code-split your app

// into multiple bundles because code-splitting a

// React Router app is the same as code-splitting

// any other React app.

export default function NestingExample() {

  return (

    <Router>

      <div>

        <ul>

          <li>

            <Link to="/">Home</Link>

          </li>

          <li>

            <Link to="/topics">Topics</Link>

          </li>

        </ul>

        <hr />

        <Switch>

          <Route exact path="/">

            <Home />

          </Route>

          <Route path="/topics">

            <Topics />

          </Route>

        </Switch>

      </div>

    </Router>

  );

}

function Home() {

  return (

    <div>

      <h2>Home</h2>

    </div>

  );

}

function Topics() {

  // The `path` lets us build <Route> paths that are

  // relative to the parent route, while the `url` lets

  // us build relative links.

  let { path, url } = useRouteMatch();

  return (

    <div>

      <h2>Topics</h2>

      <ul>

        <li>

          <Link to={`${url}/rendering`}>Rendering with React</Link>

        </li>

        <li>

          <Link to={`${url}/components`}>Components</Link>

        </li>

        <li>

          <Link to={`${url}/props-v-state`}>Props v. State</Link>

        </li>

      </ul>

      <Switch>

        <Route exact path={path}>

          <h3>Please select a topic.</h3>

        </Route>

        <Route path={`${path}/:topicId`}>

          <Topic />

        </Route>

      </Switch>

    </div>

  );

}

function Topic() {

  // The <Route> that rendered this component has a

  // path of `/topics/:topicId`. The `:topicId` portion

  // of the URL indicates a placeholder that we can

  // get from `useParams()`.

  let { topicId } = useParams();

  return (

    <div>

      <h3>{topicId}</h3>

    </div>

  );

}

# Crud Start

import React, { useEffect, useState } from "react";

const App = () => {

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

  const [oneUser, setUser] = useState(null);

  // let oneUser;

  const URL = "http://localhost:3002";

  useEffect(() => {

    fetch(`${URL}/users`)

      .then((response) => response.json())

      .then((json) => setUsers(json));

    console.log("call");

  }, []);

  const getSingleUser = (i) => {

    fetch(`${URL}/users/${i}`)

      .then((response) => response.json())

      .then((json) => setUser(json));

    // console.log(i);

  };

  const delSingleUser = (i) => {

    fetch(`${URL}/users/${i}`, {

      method: "DELETE",

    })

      .then((response) => response.json())

      .then((json) => console.log(json));

  };

  return (

    <div>

      {oneUser && <h1>website : {oneUser.website}</h1>}

      {users.length > 0 && <h1>My Users</h1>}

      {users.length > 0 &&

        users.map((user) => (

          <ul key={user.id}>

            <li>

              <button onClick={() => getSingleUser(user.id)}>{user.id}</button>

              <button onClick={() => delSingleUser(user.id)}>

                Del:{user.id}

              </button>

              Name:{user.name} , website: {user.website}

            </li>

          </ul>

        ))}

    </div>

  );

};

export default App;

// id, name, username, email, address, phone, website, company