TP REACT-REDAX TOOLKIT

On souhaite créer un formulaire en utilisant React JS et redux toolkit qui gère la liste [des voitures](https://www.footmercato.net/selection/maroc/effectif/)

1. Créer un slice sliceVoiture.js qui contient état initiale suivante

const initState = {

    db:[{id:1, marque :"BMW"},

    {id:2, marque :"GOLF 8"},

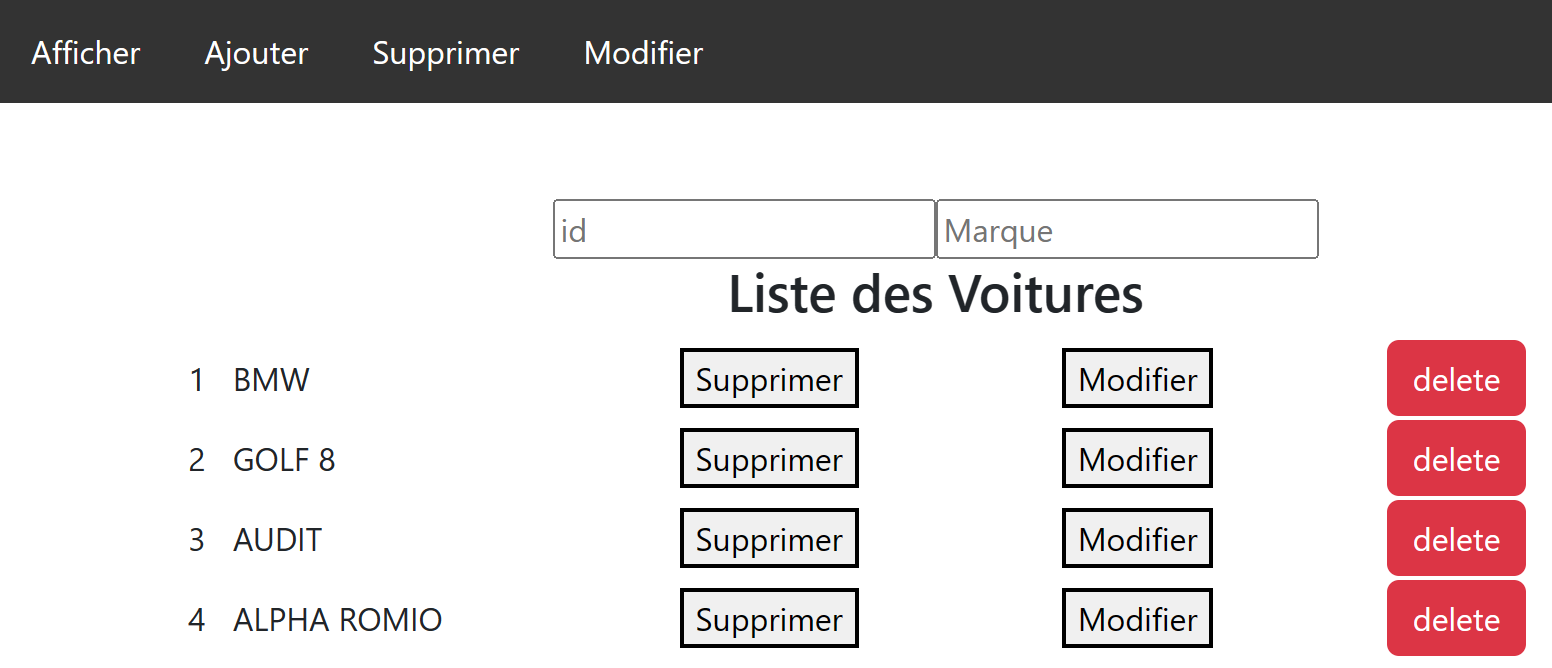
    {id:3, marque :"AUDIT"},

  {id:4, marque :"ALPHA ROMIO"}]

}

1. Ajouter au sliceVoiture quatre réducteurs permettant l’accès aux données de notre état initiale d’ ajouter, de supprimer et modifier ,recherche une voiture de la liste
2. Importer dans le store redux le slice sliceVoiture
3. Créer un composant react qui permet un d’afficher la liste des voitures (utiliser les Router (Routeur) pour accéder au différents composants de ce TP)

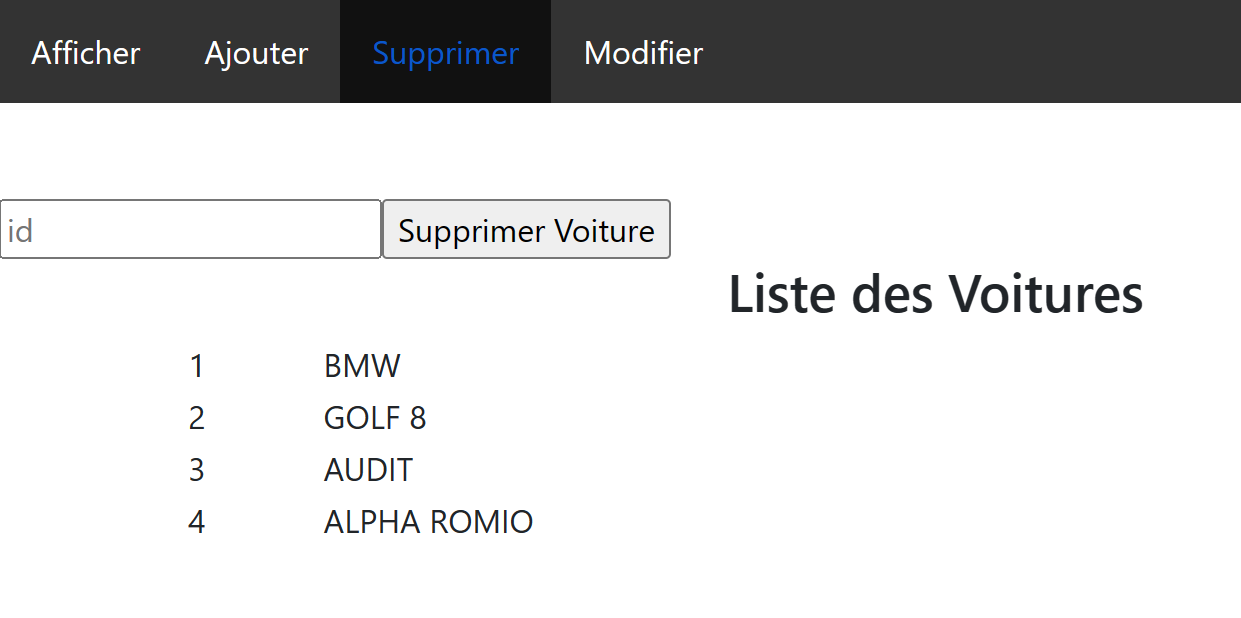
* Bouton supprimer permet de supprimer voiture en cours
* Bouton delete permet de supprimer voiture en cours avec confirmation
* Modifier permet de modifier voiture en cours après le saisi de marque



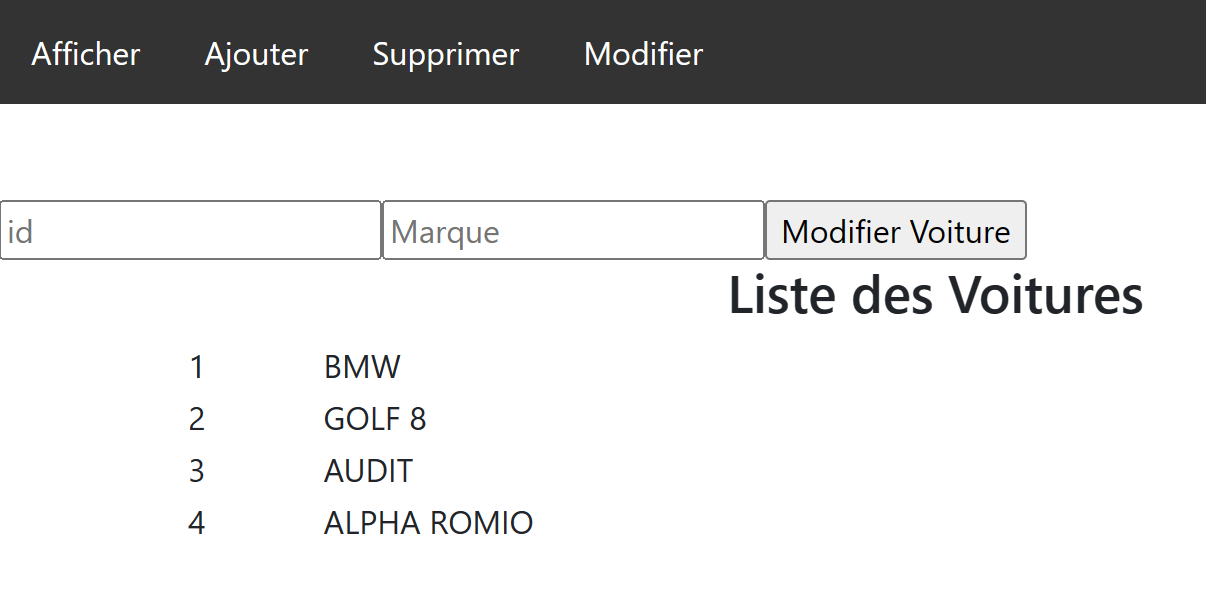
1. Ajouter Un composant qui permet ajouter une voiture



1. Créer un autre composant qui permet de de supprimer par id une voiture



1. Créer un autre composant qui permet de de modifier une voiture par id



Correction :

***voituresSlice.js***

import { createSlice } from "@reduxjs/toolkit";

const initState = {

    db:[{id:1, marque :"BMW"},

    {id:2, marque :"GOLF 8"},

    {id:3, marque :"AUDIT"},

    {id:4, marque :"ALPHA ROMIO"}]

}

const voituresSlice = createSlice({

    name : "voitures",

    initialState : initState,

    reducers:{

        addVoitures: (state, action)=>{ state.db = [...state.db, action.payload]

            console.log(action.payload); },

        RemoveVoitures: (state, action)=>{ //state.db.splice(action.payload,1)

            state.db = state.db.filter((v) => v.id !== action.payload)

        console.log(action.payload);},

       UpdateVoitures: (state, action)=>{ //state.db.splice(action.payload,1)},

        state.db.map(voitu => {if(voitu.id==action.payload.id) voitu.marque = action.payload.marque;})

         },

    }

})

export default voituresSlice.reducer;

export const {addVoitures, RemoveVoitures, UpdateVoitures,UpdateVoitures2} = voituresSlice.actions;

***strore.js***

import { configureStore } from "@reduxjs/toolkit";

import voituresSlice from "./voituresSlice";

const store =configureStore({

    reducer :{

        voitures : voituresSlice

    }

})

export default store;

***APP.JS***

import React, { Component } from 'react';

import './App.css';

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

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

import { useState } from "react";

import { useDispatch, useSelector } from "react-redux";

import { addVoitures , RemoveVoitures,   UpdateVoitures} from "./voituresSlice";

function App() {

    const voitures = useSelector(st => st.voitures.db);

    const [voitureId, setVoitureId] = useState();

    const [voitureMarque, setVoitureMarque] = useState();

    const dispatch = useDispatch();

    return (

        <BrowserRouter>   <div>

<div>

<nav>

<ul>

<li>

<Link to="/">Afficher</Link>

</li>

<li>

<Link to="/ajouter">Ajouter</Link>

</li>

<li>

<Link to="/supprimer">Supprimer</Link>

</li>

<li>

<Link to="/modifier">Modifier</Link>

</li>

</ul>

</nav>

<br></br><br></br>

<div className="main-route-place">

<Routes>

<Route path="/" element={<Afficher />} />

<Route path="/ajouter" element={<Ajouter />} />

<Route path="/supprimer" element={<Supprimer />} />

<Route path="/modifier" element={<Modifier />} />

</Routes>

</div>

</div>

          <br></br><br></br>

        </div>        </BrowserRouter>

     );

}

const Afficher = () => {

    const voitures = useSelector(st => st.voitures.db);

    const dispatch = useDispatch();

    const [voitureId, setVoitureId] = useState();

    const [voitureMarque, setVoitureMarque] = useState();

    const handleDelete = (id) =>{

        window.confirm('vous êtes sûr de supprimer ?') && dispatch(RemoveVoitures(id))

    }

    return <center>

      <input type="text" placeholder="id" value={voitureId} onChange={(e)=>setVoitureId(e.target.value)}  />

    <input type="text" placeholder="Marque" value={voitureMarque} onChange={(e)=>setVoitureMarque(e.target.value)}  /> <h3>Liste des Voitures</h3> <table border="0" width="80%">

    {

       voitures.map(voitu => <tr key={voitu.id}><td>{voitu.id }</td><td>{voitu.marque }</td><td>

       <button  onClick={()=>dispatch(RemoveVoitures(voitu.id))}>Supprimer</button></td><td>

        <button onClick={()=>{dispatch(UpdateVoitures( {id:voitu.id, marque :voitureMarque}))    ; setVoitureMarque('');}}>Modifier</button>    </td><td>

          <button className="btn btn-danger" onClick={()=>handleDelete(voitu.id)} >delete</button></td></tr>)

    }

    </table></center> ;

  };

  const Ajouter = () => {

    const [voitureId, setVoitureId] = useState();

    const [voitureMarque, setVoitureMarque] = useState();

    const voitures = useSelector(st => st.voitures.db);

    const dispatch = useDispatch();

    const handleAdd= (e)=>{

        e.preventDefault();

        dispatch(addVoitures({id:voitureId, marque :voitureMarque}))

        setVoitureId('');

        setVoitureMarque('');

    }

    return  <div>

    <form onSubmit={handleAdd}>

    <input type="text" placeholder="id" value={voitureId} onChange={(e)=>setVoitureId(e.target.value)}  />

    <input type="text" placeholder="Marque" value={voitureMarque} onChange={(e)=>setVoitureMarque(e.target.value)}  />

    <input type="submit" value="Ajouter Voiture"/>

    </form>

</div>;

  };

  const Supprimer = () => {

    const voitures = useSelector(st => st.voitures.db);

    const dispatch = useDispatch();

    const [voitureId, setVoitureId] = useState();

    const handledelete= (e)=>{

        e.preventDefault();

        dispatch(RemoveVoitures({id:voitureId}))

        setVoitureId('');

    }

    return <div>

    <form onSubmit={handledelete}>

    <input type="text" placeholder="id" value={voitureId} onChange={(e)=>setVoitureId(e.target.value)}  />

    <input type="submit" value="Supprimer Voiture"/>

    <center> <h3>Liste des Voitures</h3> <table border="0" width="80%">

    { voitures.map(voitu => <tr key={voitu.id}><td>{voitu.id }</td><td>{voitu.marque }</td></tr>)

  }

    </table></center>  </form></div>;

  };

  const Modifier = () => {

    const voitures = useSelector(st => st.voitures.db);

    const [voitureMarque, setVoitureMarque] = useState();

    const dispatch = useDispatch();

    const [voitureId, setVoitureId] = useState();

    const handleUpdate= (e)=>{

      e.preventDefault();

      dispatch(UpdateVoitures({id:voitureId, marque :voitureMarque}))

      setVoitureId('');

      setVoitureMarque('');

  }

    return <div>

    <form onSubmit={handleUpdate}>

    <input type="text" placeholder="id" value={voitureId} onChange={(e)=>setVoitureId(e.target.value)}  />

    <input type="text" placeholder="Marque" value={voitureMarque} onChange={(e)=>setVoitureMarque(e.target.value)}  />

    <input type="submit" value="Modifier Voiture"/>

    <center> <h3>Liste des Voitures</h3> <table border="0" width="80%">

    { voitures.map(voitu => <tr key={voitu.id}><td>{voitu.id }</td><td>{voitu.marque }</td></tr>)

  }

    </table></center>  </form></div>;

  };

export default App;

***INDEX.JS***

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import { Provider } from 'react-redux';

import store from './store';

import reportWebVitals from './reportWebVitals';

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

root.render(

  <Provider store={store}>

  <App />

</Provider>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

***APP. CSS***

.App {

  text-align: center;

}

.App-logo {

  height: 40vmin;

  pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

  .App-logo {

    animation: App-logo-spin infinite 20s linear;

  }

}

.App-header {

  background-color: #282c34;

  min-height: 100vh;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  font-size: calc(10px + 2vmin);

  color: white;

}

.App-link {

  color: #61dafb;

}

@keyframes App-logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

header{

  width: 100%;

  height: 100px;

  display: flex;

  justify-items: center;

  justify-content: space-around;

}

header ul{

  display: flex;

}

header ul li {

  list-style-type: none;

  margin: 20px;

}

ul {

  list-style-type: none;

  margin: 0;

  padding: 0;

  overflow: hidden;

  background-color: #333;

}

li {

  float: left;

}

li a {

  display: block;

  color: white;

  text-align: center;

  padding: 14px 16px;

  text-decoration: none;

}

li a:hover {

  background-color: #111;

}