

Le but de cet exercice est de vous faire réaliser une application composée :

- 1- D'un store redux permettant de gérer l'état global de l'application : un ensemble de pays.
 - ✓ Le store a un état initial contenant un tableau avec les données sur les pays. Pour chaque pays, on doit disposer des informations suivantes :
 - Code : code du pays,
 - Name : nom du pays,
 - Continent : continent,
 - SurfaceArea : superficie,
 - Image : url
 - Indep Year : année d'indépendance (si absent : pays dépendant d'un autre),
 - Population : population totale,
 - Cities : tableau décrivant les villes du pays connues dans la base
 - Name : nom de la capitale,
 - District : province/région,
 - Population : population de la ville,
 - Capital : true si c'est la capitale

Et de(s) reducer(s) permettant l'accès aux données, l'ajout d'une ville ou la modification de la population d'un pays

- ✓ Des composants react qui accèdent au store pour afficher les données, les ajouter, les modifier ou les filtrer.

Travail à faire

- 1- Créer le store redux décrit ci-dessus avec son état, ses reducers et ses actions creators. **(6pts)**
- 2- Créer un composant « Menu » pour accéder aux différents composants. **(4pts)**
- 3- Créer un composant « Q3 » qui affiche tous les pays sous forme des cartes « Cards » contenant l'image, le nom, la superficie et la population d'un pays. **(6pts)**
- 4- Créer un composant « Q4 » qui affiche les années de l'indépendance sous forme des liens hypertextes au clic sur un lien. On affiche dans le composant Q3 les pays qui ont obtenu l'indépendance dans l'année choisie. **(6pts)**
- 5- Créer un composant « Q5 » permet d'ajouter une ville ou de modifier sa population. **(6pts)**
- 6- Créer un composant « Q6 » qui permet de filtrer les pays par population ou par continent (le choix du continent se fait via une liste déroulante). **(6pts)**
- 7- Soit " <https://restcountries.com/v3.1/all> » l'URL de l'API qui retourne la liste de tous les pays, ajouter dans votre store une fonction asynchrone qui remplit le tableau pays de votre state par les pays fournis par l'API. **(6PTS)**

Correction

1- Question 1

Pour la correction de cet exercice on a décidé de mettre toutes les ressources dans un fichier qui s'appelle **ResourcesJSON.js** ce fichier contient :

InitialState : un tableau vide [] qui représente l'état initial.

listeContinents : Un tableau qui contient la liste des continents(question 6)

PaysParAnneeIndependence : un tableau d'objets qui contient sous cette forme [...,{

```
"country": "Morocco",  
"independence": 1956  
},...].
```

(question 7)

La création du store se fait en trois étapes :

- 1- Création des actions
- 2- Création du reducer
- 3- Création du store

Store.js

```
import { createStore } from "redux";  
import { InitialState, PaysParAnneeIndependence } from  
"./ResourcesJSON";  
//1 - Création des actions  
export const AjouterVille=(p)=>{return  
{type:"AjouterVille",payload:p}}  
export const ModifierPopulation=(p)=>{return  
{type:"ModifierPopulation",payload:p}}  
  
//2- Création du reducer  
export const rdcr=(state=InitialState,action)=>{  
  if(action.type=="AjouterVille") {  
  
    return state.map(el=> el.Code===action.payload.Code?  
      {...el  
        ,Villes:[...el.Villes,action.payload.Ville]  
      }: el);  
  }  
  else if(action.type=="ModifierPopulation") {  
  
    return state.map(el=>el.Code===action.payload.Code?  
      {...el,Population:action.payload.Population}:el );  
  }  
}
```

```

    }

    else if(action.type=="AjouterPays") {
        return [...state,action.payload];
    }

    else return state;
}
//3- Création du store
export const PaysStore=createStore(rdcr);

```

2- Menu

Ici on va utiliser NavLink pour créer des liens, car NavLink contient par défaut une classe css qui s'appelle 'active', qui donne un style à l'élément de menu actif(cliqué), on peut réécrire le code de cette classe dans le fichier index.css comme suit :

```

.active{
    background-color: antiquewhite;
}

```

Menu.js

```

import React from 'react'
import { Link, NavLink, Outlet } from 'react-router-dom'
export default function Menu() {
    return ( <>
        <nav>
            <NavLink to="/Q3" > Q3 </NavLink>|
            <NavLink to="/Q4" > Q4 </NavLink>|
            <NavLink to="/Q5" > Q5 </NavLink>|
            <NavLink to="/Q6" > Q6 </NavLink>|

        </nav>
        <Outlet/>
    </>
)
}

```

Résultat

[Q3](#) | [Q4](#) | [Q5](#) | [Q6](#) |

Index.js

```
import './index.css';

import { BrowserRouter, Route, Routes } from 'react-router-dom';
import ReactDOM from 'react-dom';
import Q3 from './Q3';
import Q5 from './Q5';
import Q6 from './Q6';
import Q4 from './Q4';
import Menu from './Menu';
import { Provider } from 'react-redux';
import { PaysStore } from './store';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <>
    <BrowserRouter>
    <Provider store={PaysStore}>
      <Menu />

      <Routes>
        <Route path="/" element={<Q3/>} ></Route>
        <Route path="/Q3" element={<Q3/>} ></Route>
        <Route path="/Q3/:annee" element={<Q3/>} ></Route>
        <Route path="/Q4" element={<Q4/>} ></Route>
        <Route path="/Q5" element={<Q5/>} ></Route>
        <Route path="/Q6" element={<Q6/>} ></Route>

      </Routes>
    </Provider>
  </BrowserRouter>

  </>
);
```

3- Q3.js

- on utilise le hook useParams pour savoir si on veut afficher la liste entière des pays, ou filtrer la liste selon l'année de l'indépendance (voir question 4).
- Le bouton 'Charger liste Pays' fait le chargement asynchrone de la liste des pays depuis une API(voir question 7).

```
import React from 'react';
import { useSelector } from 'react-redux';
import { useParams } from 'react-router-dom';
import { AjouterPays_Async, AjouterVille, ModifierPopulation,
```

```
PaysStore } from './store'

export default function Q3() {

  const {annee}=useParams();
  const select=useSelector(state=> annee?
    state.filter(el=>el.IndepYear==annee):state);
  return (<>
    { /* <div>Q3</div> */ }
    {annee && "Année Indépendance:" + annee}
    <div className="row">
      {select.map(element=> <div key={element.Code} className="card"
        style={{width: "18rem"}}>
        <img src={element.Image} className="card-img-top"
        alt={element.Name}/>
        <div className="card-body">
          <h5 className="card-title">{element.Name}</h5>
          <p className="card-text">Superficie: {element.SurfaceArea}
            km²</p>
          <p className="card-text">Population: {element.Population}</p>
        </div>
        </div>
      )}</div>

    <button onClick={()=>PaysStore.dispatch(AjouterPays_Async())}>Charger
    liste Pays</button>

    </>)
  }
```

Résultat

[Q3](#) | [Q4](#) | [Q5](#) | [Q6](#)



Saint Vincent and the Grenadines
Superficie: 389 km²
Population: 110947



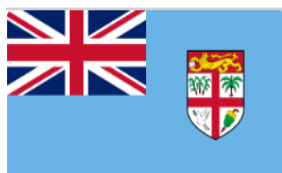
French Guiana
Superficie: 83534 km²
Population: 254541



Faroe Islands
Superficie: 1393 km²
Population: 48865



Pakistan
Superficie: 881912 km²
Population: 220892331



Fiji
Superficie: 18272 km²



Mongolia
Superficie: 1564110 km²



Cocos (Keeling) Islands
Superficie: 14 km²



Micronesia
Superficie: 702 km²

4- Q4.js

On va charger la liste des années d'indépendance depuis le store, puis éliminer les années en double, et enfin les classer par ordre croissant.

La liste des dates sera le tableau à afficher en utilisant la fonction map.

```
import React from 'react'
import { useSelector } from 'react-redux'
import { Link } from 'react-router-dom';

export default function Q4() {

  const
  listeDates=useSelector(st=>st.map(e=>e.IndepYear).filter(el=>el));
  const listeDatesUniques=[];
  listeDates.forEach(element => {
    if(listeDatesUniques.indexOf(element)==-1)
      listeDatesUniques.push(element);
  });
  listeDatesUniques.sort();
  return (<>
    /* <div>Q4</div> */
    <ul class="list-group">
    {listeDatesUniques.map(el=> <li class="list-group-item">
      <Link to={"/Q3/"+el}>{el}</Link>
    </li>
    )}
    </ul>
    </>)
  }
```

Résultat

Q3 Q4 Q5 Q6
-1000
-1523
-660
1066
1143
1278
1350
1492
1499
1581
1769
1776

5- Q5.js

```
import React, { useState } from 'react'
import { useSelector } from 'react-redux';
import { AjouterVille, ModifierPopulation, PaysStore } from
'./store';

export default function Q5() {
  const [PaysChoisi,setPaysChoisi]=useState("");
  const [ville,setVille]=useState("");
  const listePays=
useSelector(state=>state.map(country=>{return  {"Code":country.Code,"
Name":country.Name}}));

  return (
    <>
    <select onChange={(ev)=>setPaysChoisi(st=>ev.target.value)}>
{listePays.map(e=> <option value={e.Code} >{e.Name}</option> )}
</select>{PaysChoisi}<br/>
    <input name="Name" placeholder='Name'
onChange={(ev)=>setVille({...ville,Name:ev.target.value})}/><br/>
    <input name="District" placeholder='District'
onChange={(ev)=>setVille({...ville,District:ev.target.value})}/><br/>
    <input name="Population" placeholder='Population'
onChange={(ev)=>setVille({...ville,Population:ev.target.value})}/><br
/>
    Capital<input type="checkbox" name="Capital"
placeholder='Capital'
onChange={(ev)=>setVille({...ville,Capital:ev.target.checked})}/><br/
    >
    {JSON.stringify(ville)}
    <button className='btn btn-primary' onClick={()=>{
      PaysStore.dispatch(AjouterVille({Code:PaysChoisi,Ville:ville}
))};}
    >>Ajouter Ville</button>

    <button className='btn btn-info' onClick={()=>{
      PaysStore.dispatch(ModifierPopulation({Code:PaysChoisi,Popula
tion:ville.Population}));}
    >>Modifier Population</button>
    </>
  )
}
```

Résultat

[Q3](#) | [Q4](#) | [Q5](#) | [Q6](#) |

Saint Vincent and the Grenadines

▼

Name
District
Population

Capital ☐

Ajouter Ville

Modifier Population

6- Q6.js

```
import React, { useState } from "react";
import { useSelector } from "react-redux";
import { listeContinents } from "../ResourcesJSON";

export default function Q6() {
  const [continent, setContinent] = useState("");
  const [population, setPopulation] = useState("");
  const paysParContinent = useSelector((st) =>
    st.filter(
      (el) =>
        (continent ? el.Continent == continent : true) &&
        (population ? el.Population > population : true)
    )
  );

  return (
    <>
      { /* <div>Q6</div> */ }
      <input
        type="text"
        placeholder="population >"
        onChange={(ev) => setPopulation((c) => ev.target.value)}
      />
      <select onChange={(ev) => setContinent((c) =>
        ev.target.value)}>
        {listeContinents.map((e) => (
          <option value={e}>{e}</option>
        ))}
      </select>
    </>
  );
}
```


Durée : 2 h30
Filière : Développement Digital-option web full stack
Année : 2022 /2023

Control Continu 2 (Note sur 40)
Niveau : TS
Formateur : Khalid MZIBRA

```









</select>
<div className="row">
  {paysParContinent.map((element) => (
    <div key={element.Code} className="card" style={{ width:
"18rem" }}>
      <img
        src={element.Image}
        className="card-img-top"
        alt={element.Name}
      />
      <div className="card-body">
        <h5 className="card-title">{element.Name}</h5>
        <p className="card-text">Superficie:
{element.SurfaceArea} km²</p>
        <p className="card-text">Population:
{element.Population}</p>
        <a href="#" className="btn btn-primary">
          Go somewhere
        </a>
      </div>
    </div>
  ))}
</div>
</>
);
}

```

Résultat

Q3 | Q4 | Q5 | Q6 |

36000000 Africa

 <p>DR Congo Superficie: 2344858 km² Population: 108407721 Go somewhere</p>	 <p>South Africa Superficie: 1221037 km² Population: 59308690 Go somewhere</p>	 <p>Morocco Superficie: 446550 km² Population: 36910558 Go somewhere</p>	 <p>Tanzania Superficie: 945087 km² Population: 59734213 Go somewhere</p>
 <p>Nigeria Superficie: 923768 km²</p>		 <p>Ethiopia Superficie: 1104300 km²</p>	 <p>Sudan Superficie: 1886068 km²</p>

7-

on aura besoin d'installer la bibliothèque axios afin d'extraire les données qui existent dans un site distant.

L'utilisation de redux-thunk est imposé car on va ajouter des actions asynchrones.

Les étapes à respecter sont :

- 1- installer redux-thunk et axios (npm install axios redux-thunk)
- 2- Importer axios et thunk au niveau de store.js

```
import axios from 'axios';
import thunk from "redux-thunk";
```

- 3- Importer applyMiddleware depuis 'redux'
- 4- Créer une action asynchrone `AjouterPays_Async` qui se charge d'importer la liste pays depuis l'API, pour chaque pays on exécute une action qui l'ajoute au store `AjouterPays`
- 5- Ajouter le bouton qui fait le dispatch de l'action au niveau du composant Q3

```
<button onClick={()=>PaysStore.dispatch(AjouterPays_Async())}>Charger
liste Pays</button>
```

```
import {applyMiddleware, createStore} from "redux";
import axios from 'axios';

export const AjouterPays_Async=(()=>{
  return (dispatch,state)=>{
    axios.get("https://restcountries.com/v3.1/all")
      .then(res=> {
        res.data.forEach(elem=>{
          const
IndepPays=PaysParAnneeIndependence.filter(p=>p.country==elem.name.common
)[0];

          const Pays={"Code":elem.cca2,"Name":elem.name.common
,"Continent":elem.region,"SurfaceArea":elem.area
,"Image":elem.flags.png,"Population":elem.population
,"IndepYear":IndepPays ? IndepPays.independence:'',
"Villes":[]}
          dispatch(AjouterPays(Pays))
        })
      })
  }
})

export const PaysStore=createStore(rdc,applyMiddleware(thunk));
```

Durée : 2 h30
Filière : Développement Digital-option web full stack
Année : 2022 /2023

Control Continu 2 (Note sur 40)
Niveau : TS
Formateur : Khalid MZIBRA

```
//Q3.js
```

```
<button onClick={()=>PaysStore.dispatch(AjouterPays_Async())}>Charger  
liste Pays</button>
```

Code Source du Projet

<https://github.com/mzibrakhalid/efm-reg-casa-react.git>