



FRONTEND PROGRAM

3- Redux

Redux

What is Redux?

Redux est une bibliothèque de gestion d'état populaire pour les applications JavaScript, en particulier celles construites avec React. Elle offre un moyen prévisible et centralisé de gérer l'état de l'application, ce qui facilite le suivi des données et de leurs mises à jour dans les différents composants.

1- The reducer :

The reducer is nothing but a pure function that takes `currentState` and `Action` and returns a new state. A valid Reducer can return the current state.

```
const counterSlice = createSlice({
  name: "counter",
  initialState,
  reducers: {
    increment: (state) => {
      state.value += 1;
    },
    decrement: (state) => {
      state.value -= 1;
    },
    incrementByAmount: (state, action: PayloadAction<number>) => {
      state.value += action.payload;
    },
  },
  extraReducers: (builder) => {
    builder
      .addCase(incrementAsync.pending, () => {
        console.log("incrementAsync.pending");
      })
      .addCase(
        incrementAsync.fulfilled,
        (state, action: PayloadAction<number>) => {
          state.value += action.payload;
        }
      );
  },
});
```

la création de slices (tranches) de Redux, encapsulant les reducers, les actions et l'état initial.

2- Store :

le store est un objet centralisé qui détient l'état global de votre application

```
1  import { configureStore } from "@reduxjs/toolkit";  
2  
3  import counterReducer from "../counter/counterSlice";  
4  
5  export const store = configureStore({  
6    reducer: {  
7      counter: counterReducer,  
8    },  
9  });  
10  
11  export type RootState = ReturnType<typeof store.getState>;  
12  export type AppDispatch = typeof store.dispatch;  
13
```

3-

Use react redux provider pour connecter notre redux toolkit avec notre application :

```
import ReactDOM from "react-dom/client";  
import App from "../App.tsx";  
import "../index.css";  
import { Provider } from "react-redux";  
import { store } from "../state/store.ts";  
  
ReactDOM.createRoot(document.getElementById("root")!).render(  
  <Provider store={store}>  
    <App />  
  </Provider>  
>);
```

4-Implementation :

1. Se connecte au Redux store.
2. Lit la valeur du compteur à partir du store.
3. Fournit des boutons pour déclencher des actions d'incrément et de décrémentation.

4. Dispatches les actions appropriées au store pour modifier l'état du compteur.
5. React-Redux s'assure que le composant se re-renderise lorsque l'état du compteur change, reflétant la nouvelle valeur dans l'interface utilisateur.

```
import { AppDispatch, RootState } from "../state/store";
import {
  decrement,
  increment,
  incrementByAmount,
  incrementAsync,
} from "../state/counter/counterSlice";

const Counter = () => {
  const count = useSelector((state: RootState) => state.counter.value);
  const dispatch = useDispatch<AppDispatch>();

  return (
    <div>
      <h2>{count}</h2>
      <div>
        <button onClick={() => dispatch(incrementAsync(10))}>Increment</button>
        <button onClick={() => dispatch(decrement())}>Decrement</button>
      </div>
    </div>
  );
};

export default Counter;
```