

Gestion d'état globale

Context API, Zustand et Redux Toolkit

- [Context API](#)
- [Zustand](#)

Context API

Context API — État global natif

Intégré à React. Idéal pour des états peu fréquemment mis à jour (thème, auth, langue).

Pattern complet avec custom provider

```
// AuthContext.jsx
import { createContext, useContext, useState } from 'react';

const AuthContext = createContext(null);

export function AuthProvider({ children }) {
  const [user, setUser] = useState(null);

  const login = async (credentials) => {
    const data = await loginAPI(credentials);
    setUser(data.user);
  };

  const logout = () => setUser(null);

  return (
    <AuthContext.Provider value={{ user, login, logout }}>
      {children}
    </AuthContext.Provider>
  );
}

export function useAuth() {
  const ctx = useContext(AuthContext);
  if (!ctx) throw new Error("useAuth must be used within AuthProvider");
  return ctx;
}
```

```
// main.jsx
<AuthProvider>
  <App />
</AuthProvider>

// Composant
function Navbar() {
  const { user, logout } = useAuth();
  return <button onClick={logout}>{user?.nom} – Déconnexion</button>;
}
```

Zustand

Zustand — État global simple et performant

Bibliothèque légère (~1kb), sans boilerplate, qui évite les re-renders inutiles.

```
npm install zustand
```

Création d'un store

```
import { create } from 'zustand';

const useStore = create((set, get) => ({
  // State
  count: 0,
  user: null,

  // Actions
  increment: () => set(state => ({ count: state.count + 1 })),
  decrement: () => set(state => ({ count: state.count - 1 })),
  reset: () => set({ count: 0 }),

  setUser: (user) => set({ user }),
  logout: () => set({ user: null }),

  // Action async
  fetchUser: async (id) => {
    const user = await getUser(id);
    set({ user });
  },
}));
```

Utilisation dans les composants

```
function Compteur() {  
  // Sélection précise = re-render SEULEMENT si count change  
  const count = useStore(state => state.count);  
  const increment = useStore(state => state.increment);  
  
  return <button onClick={increment}>{count}</button>;  
}  
  
function Profil() {  
  const { user, logout } = useStore(state => ({  
    user: state.user,  
    logout: state.logout  
  }));  
}
```

Zustand vs Context API

	Context API	Zustand
Installation	Natif	npm install
Performance	Re-render de tous les consommateurs	Re-render ciblé
Boilerplate	Moyen	Minimal
DevTools	React DevTools	Redux DevTools