

# Installer Redux ( avec en supplément le middleware thunk )

Note : La librairie redux-thunk permet d'enchaîner les dispatch dans un action creator. (difficile de s'en passer )

1 - `npm install redux react-redux redux-thunk`

2 - Cherchez le fichier d'entrée de votre appli React, celui qui render votre premier composant. (Il est important que ce soit le plus haut dans la hiérarchie )

Soit le fichier d'entrée index.js :

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from "../components/app";

ReactDOM.render(
  <App />,
  document.querySelector("#root")
);
```

3 - Créez une variable pour le store Redux et lui greffer les middlewares

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from "../components/app";

// Import des Libs
import { createStore, applyMiddleware } from "redux";
import thunk from "redux-thunk";

// Création du store avec dedans le middleware thunk
const createStoreWithMiddleware = applyMiddleware(thunk)(createStore);

ReactDOM.render(
  <App />,
  document.querySelector("#root")
);
```

4 - Entourez votre composant primordial du Provider de Redux pour rendre accessible le store à vos composants ( ils auront besoin de connect quand même)

Le provider aura besoin d'avoir accès au store créé précédemment + l'ensemble de vos reducers ( donc le fichier root reducer qui les combine )

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Provider } from "react-redux";
import { createStore, applyMiddleware } from "redux";
import thunk from "redux-thunk";
import App from "../components/app";
import reducers from "../reducers";

const createStoreWithMiddleware = applyMiddleware(thunk)(createStore);
ReactDOM.render(
  <Provider
    store={createStoreWithMiddleware(
      reducers
    )}
  >
    <App />
  </Provider>,
  document.querySelector("#root")
);
```

5 - Voici à quoi ressemble un fichier root reducer vide

```
import { combineReducers } from "redux";
const rootReducer = combineReducers({
});
```

Il suffit d'y ajouter vos reducer et de leurs donner un alias. En voici un exemple.

```
import { combineReducers } from "redux";
import AuthenticationReducer from "../authentication"
import ActionInfoReducer from "../action-info"
const rootReducer = combineReducers({
  authentication: AuthenticationReducer,
});
export default rootReducer;
```

6 - ( Optionnel ) Ajouter l'extension de chrome en paramètre dans createStoreWithMiddleWare. Cela vous permettra d'avoir accès à redux dev tools et voir le store ainsi que les actions redux lancées.

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Provider } from "react-redux";
import { createStore, applyMiddleware } from "redux";
import thunk from "redux-thunk";
import App from "../components/app";
import reducers from "../reducers";

const createStoreWithMiddleware = applyMiddleware(thunk)(createStore);
ReactDOM.render(
  <Provider
    store={createStoreWithMiddleware(
      reducers,
      window.__REDUX_DEVTOOLS_EXTENSION__ &&
      window.__REDUX_DEVTOOLS_EXTENSION__()
    )}
  >
    <App />
  </Provider>,
  document.querySelector("#root")
);
```