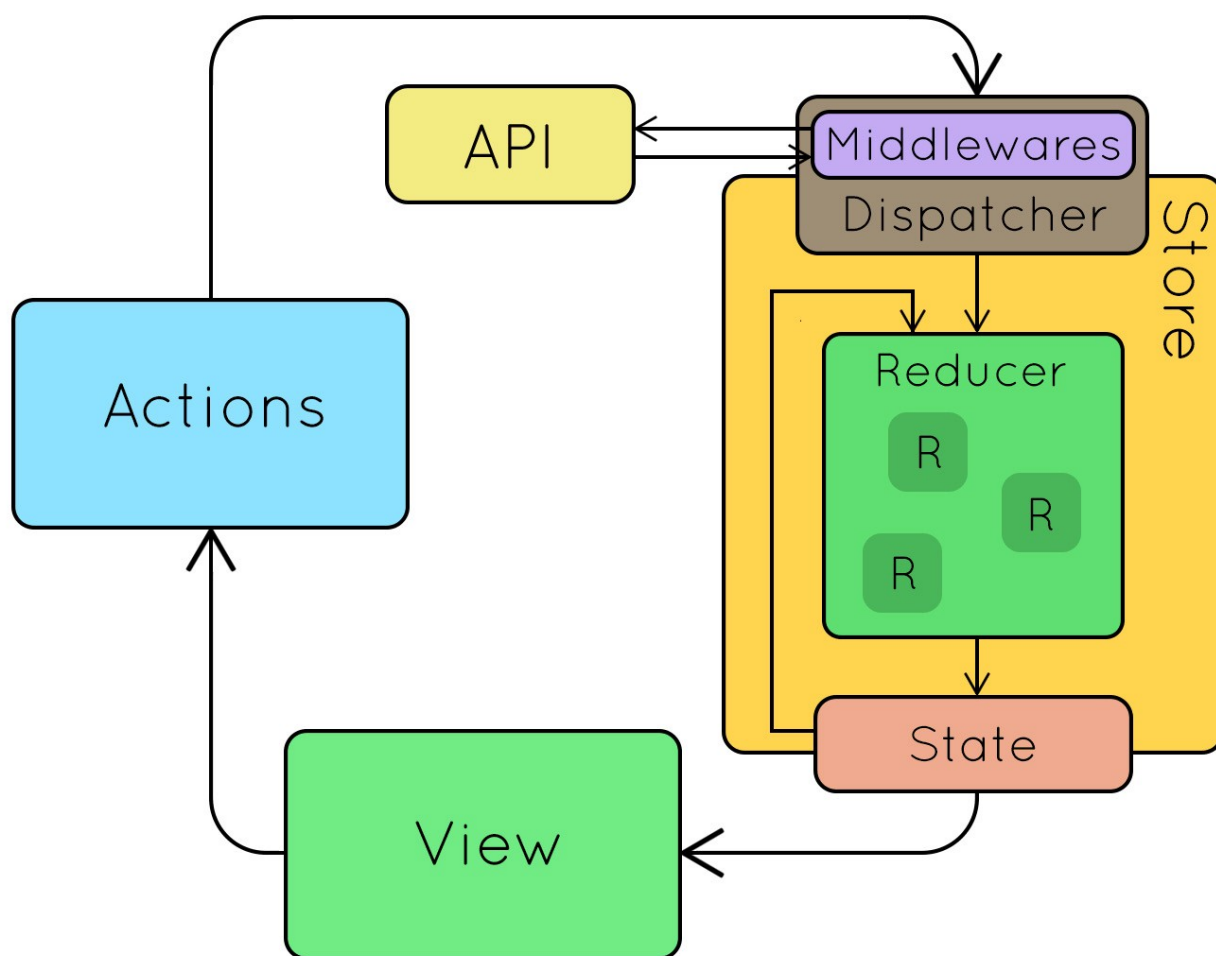


Tích hợp Redux vào ReactJS App 🎉

1. Redux trong ReactJS có khác gì trong Vanilla JS?
2. Các bước setup trong pure js app
3. Các bước setup trong ReactJS
4. Next step 🤔

1. Redux trong ReactJS có khác gì trong Vanilla JS?

- À không nhé. Dù nó ở trong app nào đây nữa thì Redux vẫn là Redux thôi nhen 🤔



2. Các bước setup trong pure js app

```
const initialState = {
  list: ['Listening to music'],
  selectedId: null,
}

const hobbyReducer = (state = initialState, action) => {
  switch (action.type) {
    case 'ADD_HOBBY': {
      const newList = [...state.list];
      newList.push(action.payload);

      return {
        ...state,
        list: newList,
      }
    }
    default:
      return state;
  }
};

const { createStore } = window.Redux;
const store = createStore(hobbyReducer);
```

3. Các bước setup trong ReactJS

1. Cài đặt package `react-redux` và `redux`

```
npm install --save redux react-redux
```

```
src
|__ reducers
|  |__ hobby.js
|  |__ todo.js
|  |__ ... (one reducer per file)
|  |__ index.js (root reducer)
|
|__ actions
|  |__ hobby.js
|  |__ todo.js
|  |__ ...
|
|__ pages
|  |__ HomePage/index.jsx (connect to redux)
|
|__ store.js (reducers, init state, middlewares)
|__ index.js (setup Store Provider)
```

2. Setup reducers và root reducer

```
// reducers/hobby.js
const initialState = {
  list: ['Listening to music'],
  selectedId: null,
}

const hobbyReducer = (state = initialState, action) => {
  switch (action.type) {
    case 'ADD_HOBBY': {
      const newList = [...state.list];
      newList.push(action.payload);

      return {
        ...state,
        list: newList,
      }
    }
    default:
      return state;
  }
};
export default hobbyReducer;
```

```
// reducers/index.js (ROOT)
const rootReducer = combineReducers({
  hobby: hobbyReducer,
})
export default rootReducer;
```

3. Setup redux store

```
// src/store.js
const store = createStore(rootReducer);
export default store;
```

4. Setup Store Provider cho toàn app `src/index.js`

```
const Main = () => (
  <Provider store={store}>
    <App />
  </Provider>
)
```

5. Connect vào redux từ reactjs component

- Với **class component**: dùng HOC **connect()**
- Với **functional component**: dùng hooks **useSelector()** và **useDispatch()**

```
function HomePage(props) {  
  const hobbyList = useSelector(state => state.hobby.list);  
  const dispatch = useDispatch();  
  
  const handleAddHobbyClick = () => {  
    const newHobby = 'Coding';  
    dispatch({  
      type: 'ADD_HOBBY',  
      payload: newHobby,  
    });  
  }  
  
  return (  
    <div className="home-page">  
      <HobbyList data={hobbyList} />  
  
      <button onClick={handleAddHobbyClick}>Add new hobby</button>  
    </div>  
  );  
}  
  
export default HomePage;
```

4. Video tiếp theo: Dùng hooks để làm việc với Redux trong ReactJS App 😊

Cảm ơn các bạn đã xem video của mình!

Nhớ like, share và subscribe để cho bạn bè cùng xem nhen 😊

❤️ Ủng hộ mình làm videos thì đóng góp tại đây nhé:

- Ủng hộ tôi: <https://unghotoi.com/easyfrontend>
- MoMo/ZaloPay: 0901 309 729

Kết nối với mình:

- 📺 Kênh Easy Frontend: <https://youtube.com/easyfrontend>
- 🎨 Fan page Facebook: <https://www.facebook.com/learn.easyfrontend>
- 🗨️ Nhóm trao đổi Facebook: <https://www.facebook.com/groups/easyfrontend>