

Giới thiệu tổng quát về Redux Toolkit 🎉

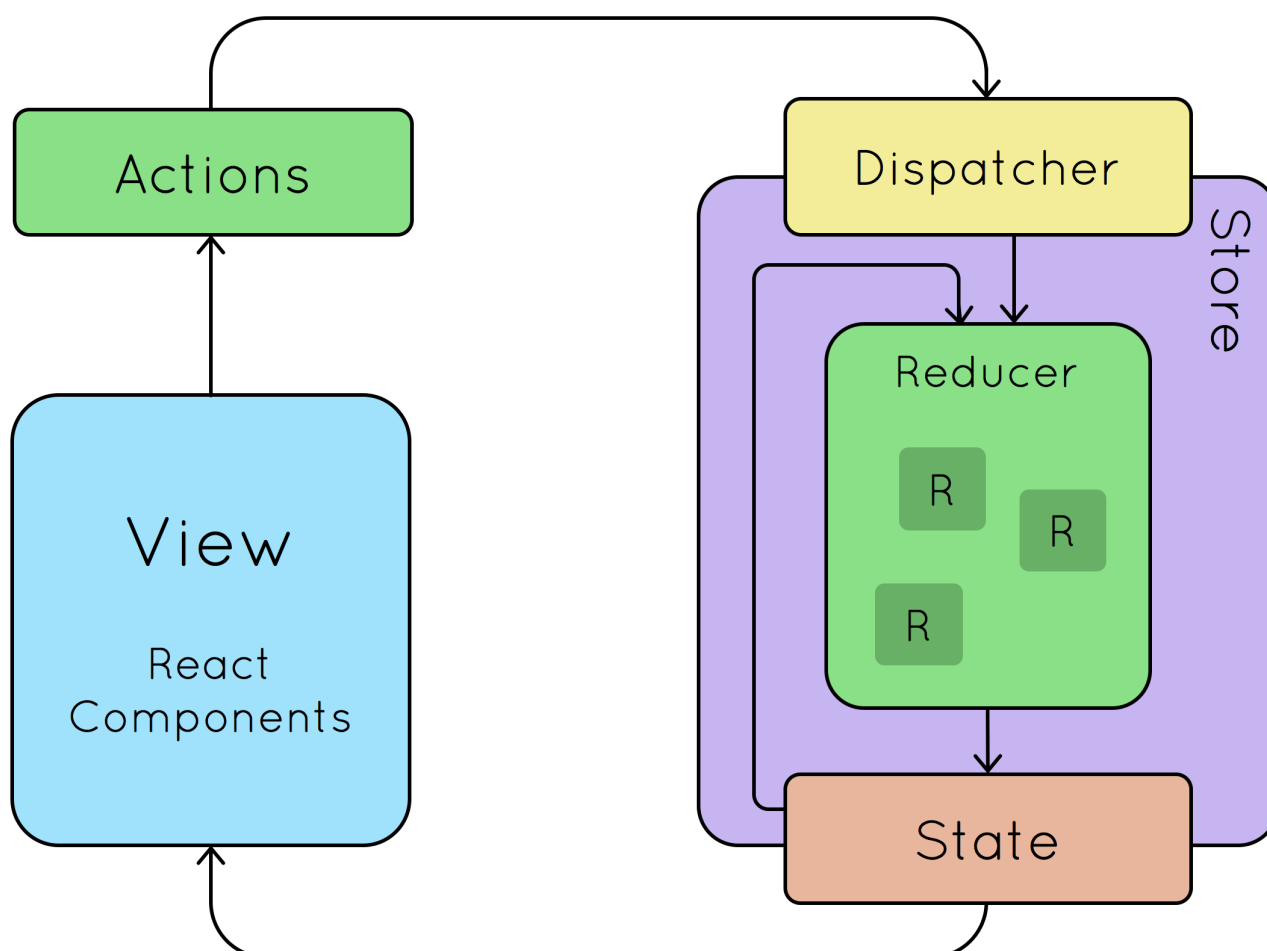
1. Redux toolkit (RTK) là gì và tại sao lại có nó?
2. RTK bao gồm những gì?
3. Setup một ví dụ đơn giản sử dụng RTK
4. Next video: Áp dụng RTK vào PhotoApp để làm tính năng ADD

Slide được biên soạn dựa trên tài liệu chính thức của Redux Toolkit:
<https://redux-toolkit.js.org/introduction/quick-start>

1. Redux Toolkit (RTK) là gì và tại sao lại có nó?

```
npm install @reduxjs/toolkit
```

- RTK là một thư viện giúp mình viết Redux tốt hơn, dễ hơn và đơn giản hơn. (tiêu chuẩn để viết Redux)
- Ba vấn đề làm nền tảng ra đời RTK:
 - Configuring a Redux store is too complicated.
 - I have to add a lot of packages to get Redux to do anything useful.
 - Redux requires too much boilerplate code.



2. RTK bao gồm những gì?

configureStore()

- Có sẵn Redux DevTools
- Có sẵn redux-thunk để thực hiện async actions

```
// Khi chưa có Redux Toolkit
// store.js
import { createStore, applyMiddleware, compose } from 'redux';
import thunkMiddleware from 'redux-thunk';
import rootReducer from './reducers';

// Enable to use redux dev tool in development mode
const composeEnhancers = 'development' === process.env.NODE_ENV
  ? (window.__REDUX_DEVTOOLS_EXTENSION_COMPOSE__ || compose)
  : compose;

// Use redux-thunk as a redux middleware
const enhancer = composeEnhancers(applyMiddleware(thunkMiddleware));

const store = createStore(rootReducer, {}, enhancer);
export default store;
```

```
// Khi đã có redux toolkit 🤖
// store.js
import { configureStore } from '@reduxjs/toolkit'
import rootReducer from './reducers'
const store = configureStore({ reducer: rootReducer })
```

createReducer()

```
// Không có Redux Toolkit
function counterReducer(state = 0, action) {
  switch (action.type) {
    case 'increment':
      return state + action.payload
    case 'decrement':
      return state - action.payload
    default:
      return state
  }
}
```

```
// Có Redux Toolkit
// - Mỗi key là một case
// - Không cần handle default case
const counterReducer = createReducer(0, {
  increment: (state, action) => state + action.payload,
  decrement: (state, action) => state - action.payload
})
```

```
// Một điểm hay nữa là reducer có thể mutate data trực tiếp.
// Bản chất bên dưới họ sử dụng thư viện Immerjs
const todoReducer = createReducer([], {
  addTodo: (state, action) => {
    // 1. Có thể mutate data trực tiếp 🚀
    state.push(action.payload)
  },
  removeTodo: (state, action) => {
    // 2. Hoặc phải trả về state mới
    // CHÚ Ý KHÔNG ĐƯỢC cả 1 và 2 nha 😏
    const newState = [...state];
    newState.splice(action.payload, 1);
    return newState;
  }
})
```

createAction()

```
// Không có redux toolkit
const INCREMENT = 'counter/increment'

function increment(amount) {
  return {
    type: INCREMENT,
    payload: amount
  }
}

const action = increment(3)
// { type: 'counter/increment', payload: 3 }
```

```
// Có redux toolkit
const increment = createAction('counter/increment')

const action = increment(3)
// returns { type: 'counter/increment', payload: 3 }

console.log(increment.toString())
// 'counter/increment'
```

Các bạn đọc thêm những hàm sau nữa nhé 😊

- `createSlice()`: <https://redux-toolkit.js.org/api/createSlice>
- `createSelector()`: <https://redux-toolkit.js.org/api/createSelector>
- `createAsyncThunk()`: <https://redux-toolkit.js.org/api/createAsyncThunk>
- `createEntityAdapter()`: <https://redux-toolkit.js.org/api/createEntityAdapter>

3. Setup một ví dụ đơn giản sử dụng RTK

```
// 1. Setup todo slice
// todoSlice.js
const todoSlice = createSlice({
  name: 'todos',
  initialState: [],
  reducers: {
    addPost(state, action) {
      state.push(action.payload);
    },
    removePost(state, action) {
      state.splice(action.payload, 1)
    }
  }
});

const { actions, reducer } = todoSlice;
export const { addPost, removePost } = actions;
export default reducer;
```

```
// 2. Setup redux store
// store.js
import { configureStore } from '@reduxjs/toolkit';
import todoSlice from 'features/todos/todoSlice';

const store = configureStore({
  reducer: {
    todos: todoSlice
  },
});
```

```
// 3. Bind Redux Provider to App
// src/index.js
import { Provider } from 'react-redux';
import store from './store';
import App from './App';

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

```
// 4. Using redux in component
// todo.jsx
import { useDispatch, useSelector } from 'react-redux';
import { removeTodo } from 'features/todos/todoSlice';

function Todo() {
  const dispatch = useDispatch();
  const todoList = useSelector(state => state.todos);

  const handleTodoClick = (todo, idx) => {
    const action = removeTodo(idx);
    dispatch(action);
  }

  return (
    <ul>
      {todoList.map((todo, idx) => (
        <li key={todo.id} onClick={() => handleTodoClick(todo, idx)}>
          {todo.title}
        </li>
      ))}
    </ul>
  )
}
```

4. Next video: Áp dụng RTK vào PhotoApp để làm tính năng ADD.

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>