

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?

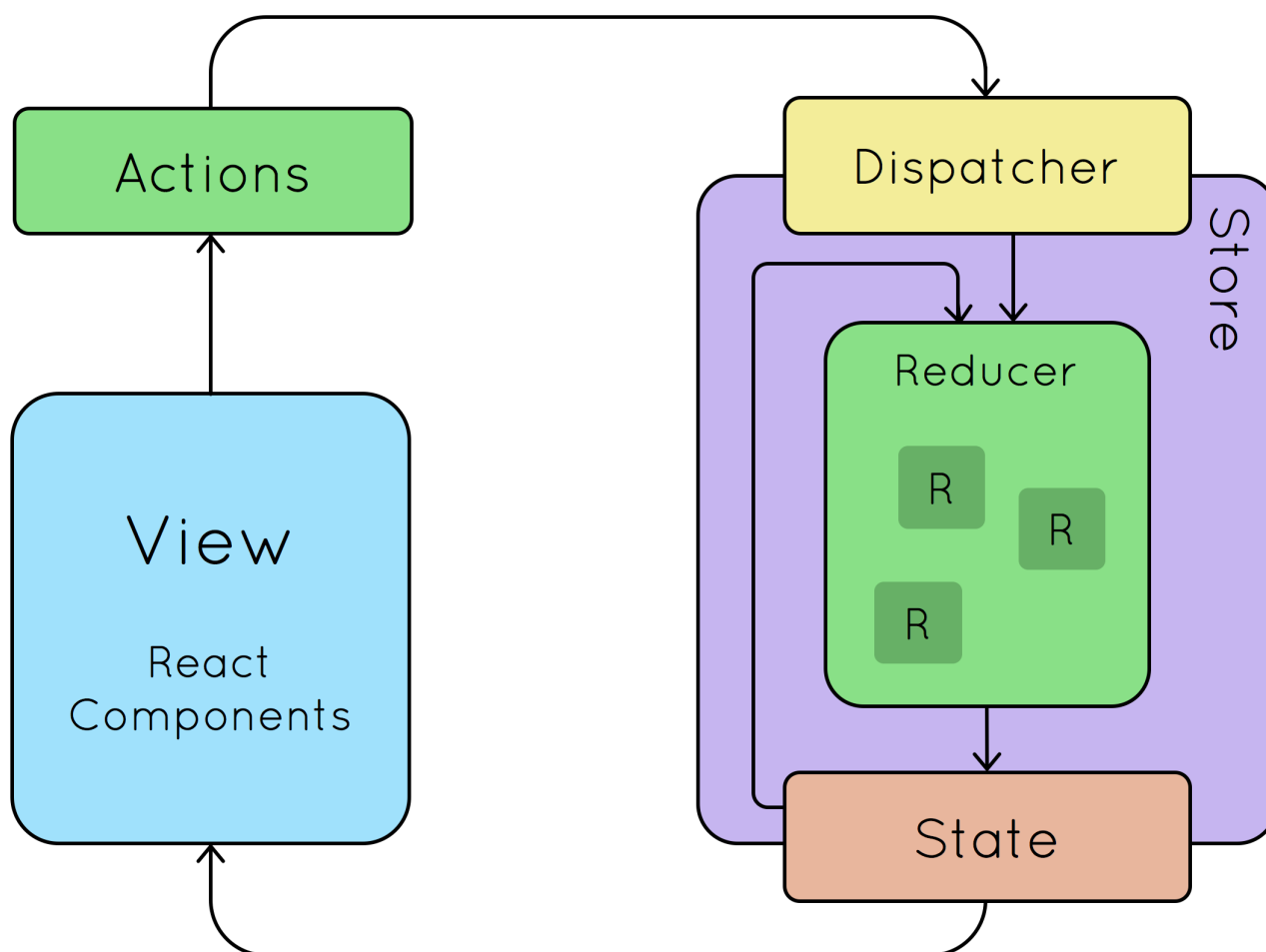
Slide được biên soạn dựa trên tài liệu chính thức của Redux Toolkit:

<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)
  },
  addTodoWithoutImmerJS: (state, action) => {
    // 2. Hoặc phải trả về state mới
    // CHÚ Ý KHÔNG ĐƯỢC cả 1 và 2 nha 😏
    const newState = [...state];
    newState.push(action.payload);
    return newState;
  }
})
```

createAction()

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

// action creator
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>