

REACTJS

Hà Nội, ngày 29 tháng 12 năm 2023

I. React Redux

I.1. Khái niệm

React Redux là một thư viện React giúp quản lý trạng thái của ứng dụng một cách hiệu quả. Nó thường được sử dụng để kết hợp React với Redux, một thư viện quản lý trạng thái dựa trên nguyên tắc Flux. React Redux giúp kết nối trạng thái của ứng dụng với các component React một cách dễ dàng và hiệu quả.

Cách cài đặt:

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

I. React Redux

Tạo Redux Store:

- Sử dụng createStore từ thư viện Redux để tạo một Redux store.

```
import { createStore } from 'redux';  
import rootReducer from './reducers'; // Định nghĩa reducer  
  
const store = createStore(rootReducer);  
  
export default store;
```



I. React Redux

Kết Hợp Redux với React:

- Sử dụng Provider từ react-redux để bao bọc gốc của ứng dụng và truyền store xuống tất cả các component.

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Provider } from 'react-redux';
import store from './store';
import App from './App';

ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>,
  document.getElementById('root')
);
```

I. React Redux

Kết Hợp Component với Redux:

- Sử dụng connect từ react-redux để kết nối component với store và map state hoặc dispatch thành props.

```
import { connect } from 'react-redux';

const MyComponent = ({ count, increment, decrement }) => {
  return (
    <div><p>Count: {count}</p>
      <button onClick={increment}>Increment</button>
    </div>
  );
};

const mapStateToProps = (state) => ({count: state.count});
const mapDispatchToProps = (dispatch) => ({
  increment: () => dispatch({ type: 'INCREMENT' }),
});
export default connect(mapStateToProps, mapDispatchToProps)(MyComponent);
```




I. React Redux

Khi nào sử dụng React Redux:

- Sử dụng React Redux khi ứng dụng có trạng thái phức tạp.
- Sử dụng khi cần chia sẻ trạng thái giữa nhiều component khác nhau mà không cần truyền dữ liệu qua props.
- Sử dụng khi muốn quản lý trạng thái tại mức ứng dụng, giúp giữ cho quản lý trạng thái trở nên dễ dàng và nhìn nhận toàn cảnh.

Cần chú ý khi sử dụng React Redux:

- Redux tuân theo nguyên tắc unidirectional data flow của Flux.
 - Tránh việc tạo quá nhiều actions. Thay vào đó, sử dụng actions mô tả các sự kiện chính trong ứng dụng.
 - Chia nhỏ reducers để giữ cho mã nguồn dễ đọc và dễ quản lý hơn.
 - Hãy cân nhắc và hạn chế sử dụng global state nếu không cần thiết để tránh sự phức tạp không mong muốn.
- 

I. React Redux

So sánh các thư viện khác:

- MobX
- React Recoil
- Zustand

So sánh với:

- Context & useContext
- useState & useReducer

THANK YOU

FOR
WATCHING