

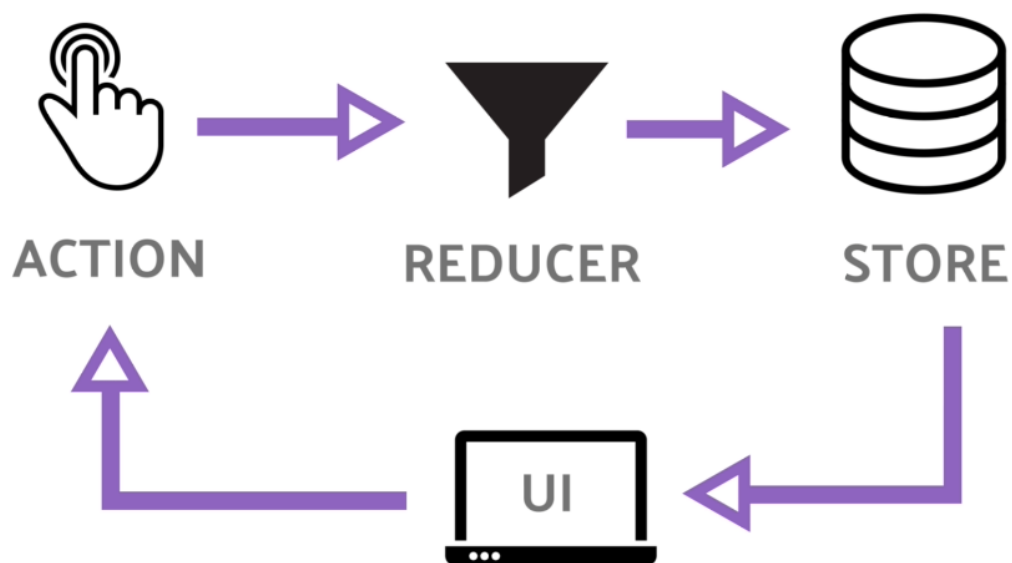
React_20220111

Redux

- JavaScript 상태 관리 라이브러리
- 본질은 Node.js 모듈

Redux의 기본 개념 : 세가지 원칙

1. Single Source of Truth
 - 동일한 데이터는 항상 같은 곳에서 가지고 온다.
 - 즉, 스토어라는 하나뿐인 데이터 공간이 있다.
2. State is read-only
 - 리액트에서는 setState 메소드를 활용해야만 상태 변경이 가능하다.
 - 리덕스에서도 액션이라는 객체를 통해서만 상태를 변경할 수 있다.
3. Changes are made with pure functions
 - 변경은 순수함수로만 가능하다.
 - 리듀서와 연관되는 개념이다.
 - Store - Action - Reducer



Store

상태가 관리되는 오직 하나의 공간

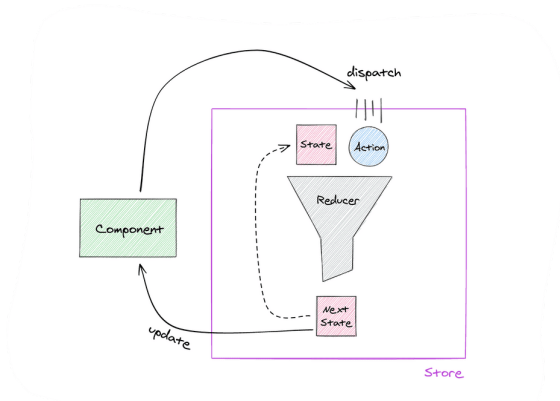
- 컴포넌트와는 별개로 스토어라는 공간이 있어서 그 스토어 안에 앱에서 필요한 상태를 담는다.
- 컴포넌트에서 상태 정보가 필요할 때 스토어에 접근한다.

Action

자바스크립트 객체 형식으로 되어있으며 앱에서 스토어에 운반할 데이터를 말한다.

Reducer

- Action을 Store가 바로 전달하는 것이 아니라 Reducer를 통해 전달한다.
- Reducer(리듀서)가 주문을 보고 Store(스토어)의 상태를 업데이트하는 것이다.
- Action(액션)을 Reducer(리듀서)에 전달하기 위해서는 `dispatch()` 메소드를 사용해야 한다.



1. Action 객체가 `dispatch` 메서드를 전달한다.
2. `dispatch`를 통해 Reducer를 호출한다.
3. Reducer는 새로운 Store를 생성한다.

데이터는 한 방향으로만 흘러야하기 때문!

Redux의 장점

- 상태를 예측 가능하게 만든다. (순수함수를 사용하기 때문)
- 유지보수 (복잡한 상태 관리와 비교)
- 디버깅에 유리 (action과 state log 기록 시) → `redux dev tool` (크롬 확장)
- 테스트를 붙이기 용의 (순수함수를 사용하기 때문)

Redux에서 위 개념을 구현하는 두 가지 방법

mapStateToProps()

Redux hooks (비교적 최근에 나온 기술)

- useSelector
- useDispatch

Ref.

<https://hanamon.kr/redux%EB%9E%80-%EB%A6%AC%EB%8D%95%EC%8A%A4-%EC%83%81%ED%83%9C-%EA%B4%80%EB%A6%AC-%EB%9D%BC%EC%9D%B4%EB%B8%8C%EB%9F%AC%EB%A6%AC/>