

## 상태관리 라이브러리

### 1. 리코일(Recoil)

- 특징
  - 리액트 문법 친화적(get/set 인터페이스로 사용할 수 있는 보일러 플레이트가 없는 API 제공)
    - ◆ 보일러 플레이트: 별 수정 없이 반복적으로 사용되는 코드
  - 비동기처리 추가적인 라이브러리 없이 리코일 안에서 가능
  - 내부적으로 캐싱 지원(동일한 atom 값에 대한 내부적으로 메모이제이션된 값 반환> 속도 빠름)
    - ◆ 아톰(상태단위): 업데이트>구독하고 있는 컴포넌트 리렌더링 발생
    - ◆ 셀렉터: 아톰 상태변화를 일으키는 순수 함수, 비동기 처리와 데이터 캐싱 기능 제공
- 단점
  - 아톰이 여러 군데에서 사용되면 부작용 발생 가능(프로젝트 규모 커질수록 상태 구조와 의존성 관리 어려워질 수 있음)
- 러닝커브: 낮음(주스탠드보다는 높음)

### 2. 주스탠드(Zustand)

- 간결한 플렉스 원칙을 바탕으로 작고 빠르게 확장 가능
- 특징
  - 특정 라이브러리에 종속되어 만들어진 도구 아님
  - 발행/구독 모델 기반
    - ◆ 스토어 상태 변경>실행할 리스너 함수 모아둠(발행)>등록된 리스너에게 상태 변경되었다 알려줌
  - 일시적 업데이트
    - ◆ 상태 자주 바뀌어도 매번 업데이트 x, 리렌더링 제어 가능
  - 탑다운 방식으로 전역 상태 접근
    - ◆ 오버뷰>디테일로 스토어 모델링 권장

- 단점
  - 성능 중요한 앱>탑다운 방식 적합x
  - 비교적 신생이라 익스텐션 적고, 공식문서도 적음
- 러닝커브: 낮음

3. MobX

4. redux